PREDICTION OF PROTEIN FUNCTION USING TEXT FEATURES EXTRACTED FROM THE BIOMEDICAL LITERATURE

by

Andrew Wong

A thesis submitted to the School of Computing

In conformity with the requirements for

the degree of Master of Science

Queen's University

Kingston, Ontario, Canada

(April, 2013)

Abstract

Proteins perform many important functions in the cell and are essential to the health of the cell and the organism. As such, there is much effort to understand the function of proteins. Due to the advances in sequencing technology, there are many sequences of proteins whose function is yet unknown. Therefore, computational systems are being developed and used to help predict protein function.

Most computational systems represent proteins using features that are derived from protein sequence or protein structure to predict function. In contrast, there are very few systems that use the biomedical literature as a source of features. Earlier work demonstrated the utility of biomedical literature as a source of text features for predicting protein subcellular location. In this thesis we build on that earlier work, and examine the effectiveness of using text features to predict *protein function*.

Using the *molecular function* and *biological process* terms from the Gene Ontology (GO) as our function classes, we trained two classifiers (k-*Nearest Neighbour* and *Support Vector Machines*) to predict protein function. The proteins were represented using text features that were extracted from biomedical abstracts based on statistical properties. For evaluation, the performance of our two classifiers was compared to that of two baseline classifiers: one that assigns function based solely on the prior distribution of protein function, and one that assigns function based on sequence similarity. The systems were trained and tested using 5-fold cross-validation over a dataset of more than 36,000 proteins.

Overall, we show that text features extracted from biomedical literature can be used to predict protein function for *any* organism. Our results also show that our text-based classifier typically has comparable performance to the sequence-similarity baseline classifier. Based on our results and what previous work had shown, we believe that text features can be integrated with other types of features to provide more accurate predictions for protein function.

Acknowledgements

I would like to express my deepest gratitude and thanks for my supervisor, Dr. Hagit Shatkay, who, with much patience and wisdom, guided me through the course of my thesis. I would also like to thank her for all the opportunities, lessons and advices she provided me with throughout my education at Queen's University.

I would like to thank my thesis examination committee, Dr. Dorothea Blostein and Dr. Sharon Regan for their insightful suggestions.

I would like to express my sincerest appreciation to my parents and my relatives, especially my mother, for their love and support.

Lastly, I would like to thank God for providing me strength to persevere through the hardest struggles during this time and for providing me with the opportunity to undertake this study. Soli Deo Gloria.

Andrew Wong

Queen's University at Kingston, Ontario

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

One of the main goals in studying proteins is to understand their function. The function of a protein refers to the role in the chemical reactions and biological processes in which it is involved. In working toward this goal, computer-based methods that can predict and annotate protein function in a high-throughput manner are actively being developed. In this thesis, we present a computational system that predicts protein function using text features extracted from the abstracts of biomedical publications that are associated with proteins.

1.1 Motivation

Proteins are macromolecules that are found within all living cells. Proteins perform many different functional roles within cells. Their cellular roles include, but are not limited to, digesting molecules, transporting molecules, signaling between cells, and providing structural support. Proteins are not only essential to the life of the cell, but the health of organisms depends upon them as well. Therefore, studying and annotating the function of proteins has become an important goal in proteomic research.

Unfortunately, annotating protein function is not a well-defined task because the definition of function varies based on context. For example, from a molecular perspective, the function of a protein refers to the chemical reaction that it is involved in whereas in a physiological context, protein function refers to the biological pathways in which the protein participates. The function of a protein can also be ambiguous because there are proteins that perform multiple functions, which are referred to as "*moonlighting proteins"* (Jeffery, 1999). For example, the protein *cytochrome c* plays a functional role in energy metabolism, and it is also

involved in the process of programmed cell death (Huberts et al. 2010). To address the ambiguity in describing protein function, several standardized methods for annotating protein functions have been suggested (Rison, et al., 2000). Of the different standards that have been proposed, the Gene Ontology (Ashburner et al. 2000) is currently the most commonly used. The Gene Ontology provides a vocabulary of terms that is used to describe different aspects of protein function. It is separated into three sub-ontologies. *molecular function*, *biological processes*, and *cellular component*. The *cellular component* sub-ontology contains terms that describe subcellular location instead of function. Nevertheless, it is an important part of the Gene Ontology because the location of a protein is often related to its function.

Despite having standardized terminology to describe protein functions, many sequenced proteins still lack functional annotation. This is in part due to the advances in sequencing technology, which has resulted in a rapid increase in the number of sequenced proteins. In contrast, however, experimental assays for investigating protein function take a much longer time to produce results. Consequently, the number of proteins that lack functional annotation continues to grow. As of March 2013, there were over 30,000,000 records of sequenced proteins, (including variations of the same protein found in different species), in UniProtKB/TrEMBL (UniProt Consortium, 2004), but only about 540,000 entries of manually annotated proteins in UniProtKB/SwissProt (UniProt Consortium, 2004). In other words, a significant number of sequenced proteins still do not have reliable function annotations. Given the current volume of unannotated protein sequences, it is impossible for experimental methods alone to provide functional annotation in a reasonable time. Therefore, computer-based function prediction systems are needed to assist in predicting protein function to aid researchers in directing future experiments.

The most common approach to predict protein function is homology-based transfer. Given a protein with an unknown function, the strategy is to identify homologs with known function in order to transfer the annotated function of the homologs to the yet-unannotated protein. The assumption underlying this approach is that if two proteins are homologous to one another, they have a common evolutionary origin, and their respective function has been conserved (Loewenstein et al., 2009).

One method to find potential homologs is to identify proteins that have highly similar sequences. This can be done computationally using alignment programs such as PSI-BLAST (Altschul et al., 1997) and HMMER (Finn et al., 2011) to compare the sequence of the unannotated proteins with sequences of proteins whose function is known and curated in databases such as UniProtKB (UniProt Consortium, 2004). GOblet (Groth et al. 2004), OntoBlast (Zehetner et al. 2003), GOtcha (Martin et al. 2004) are all examples of computer systems that use sequence alignment and homology-based transfer to annotate protein function.

However, even though sequence-based homology transfer is widely used, it is not always accurate. It has been shown that high sequence similarity does not always guarantee preserved functionality (Rost, 2002). Similarly, Aloy et al. (2003) found that even proteins that had high sequence similarity had distinct structures and intermolecular interactions.

Ultimately, it is the structure of a protein that determines the molecules with which it can interact and the reactions in which it can participate. As such, when structure data is available, structural similarity can also be used to infer homology and function. Structure similarity can be queried by using alignment programs such as FATCAT (Ye and Godzik, 2004), FAST (Zhu and Weng, 2005) and CATHEDRAL (Redfern et al. 2007) to search the Protein Data Bank, an online repository of known 3-D structures of proteins. ConFunc (Wass and Sternberg, 2008),

PHUNCTIONER (Pazos and Sternberg, 2004), and FINDSITE (Skolnick and Brylinski, 2009) are examples of systems that use structure similarity to infer homology and annotate protein function.

Unfortunately, even though protein structures are more informative indicators of function than protein sequences, they are also prone to the same problem. Proteins that share a similar structure can still have totally different functions (Bartlett et al. 2005) while proteins that have the same function may have completely different structures (Whisstock and Lesk, 2003).

In addition to the shortcomings discussed above, homology-based transfer does not work well for unannotated proteins whose sequences and structures are distinctively different from the proteins that already have functional annotations. For such instances, function prediction systems that implement machine-learning-based classifiers are more effective. There are many different machine-learning-based classification algorithms. Some examples of the commonly used classifiers are: k-*Nearest Neighbour* (Webb et al, 2011), *Neural Networks* (Bishop, 2006), *Support Vector Machines* (Webb et al, 2011) , *Naïve Bayes* (Webb et al, 2011), and *Hidden Markov Models* (Bishop, 2006). In general, machine-learning classifiers take the representation of items as input and attempt to assign class labels to each item as output. In the context of annotating protein function, the class labels assigned by the classifiers are functions, and the items being classified are proteins whose function is unknown.

In order for the classifiers to work, they first need to be trained using a training dataset. The training dataset consists of proteins that are represented using certain salient features. The features can be derived from sources such as protein sequences, protein structures, and protein interaction data. Instead of using all available information as features, it is often helpful to apply feature selection in order to identify the most informative features. Feature selection decreases the

amount of time needed to train the classifier and improves classification performance by removing from the data features that are not useful for distinguishing among different classes. After the classifiers have been trained, they can be used to assign function class labels to unannotated proteins that are also represented using the same set of features.

Function prediction systems that use machine-learning classifiers come in many varieties. Different systems vary in the features that they use to represent proteins and in the classifiers that they use to assign class labels. For example, FAAN (Clark and Radivojac, 2011) is a function prediction system that represents proteins using sequence-based features, and employs a neural network classifier for prediction. ProKnow (Pal and Eisenberg, 2005), on the other hand, uses Bayesian statistics to classify proteins and uses structural features to represent proteins. There are also computational systems that combine multiple types of features to predict function. In general, it is found that methods that use multiple data sources outperform methods that only use a single type of features (Ko and Lee, 2009). The system developed by Yao and Ruzzo (2006) integrates features derived from sequence, structure, and interaction data, and predicts function using the k-Nearest Neighbour classifier. Guan et al. (2008) also used a combination of different features in their prediction system and they used *Support Vector Machines* to classify proteins into function classes.

Even though such a diversity of computational methods already exists, there is still ongoing research for ways to improve the prediction accuracy of function prediction systems. Based on a previous work by Brady and Shatkay (2008), we know that text features extracted from published literature can be used to represent proteins. They presented a text-based classifier, EpiLoc, which can predict protein subcellular location at least as accurately as systems that use other types of features. The proteins were represented using prominent terms found in biomedical

abstracts as text features. Brady and Shatkay used PubMed to retrieve biomedical abstracts and employed a statistical test to identify terms that are over or under represented in the abstracts that are associated with the proteins. In their work, they also showed that the text-based classifier can be integrated with systems that use other types of features to improve the overall prediction accuracy (Shatkay et al. 2009). Motivated by these results, we decided to investigate the usage of text features in the context of protein function prediction. For this thesis, we introduce and evaluate a new computational system that predicts protein function using text features.

1.2 Thesis Objective and Contribution

The objective of this thesis is to evaluate the usefulness of text features in the context of protein function prediction. To this end, we implemented a protein function prediction system that uses text features extracted from the biomedical literature. We evaluated the performance of two different classifiers, *k-Nearest Neighbour* and *Support Vector Machines* using a dataset of 36,536 proteins and compared it against two other baseline classifiers to determine the effectiveness of text features for predicting protein function.

1. We adopted and modified the framework from EpiLoc to extract text features from biomedical literature for the purpose of predicting protein function. In order to extract text features that are correlated with distinct functions, we defined function class labels using terms from the Gene Ontology. We also compiled a new dataset of proteins from *UniProtKB/Swiss-Prot* and formed a text corpus by retrieving associated abstracts from *PubMed* that is suitable for protein function prediction.
2. We implemented a *k-Nearest Neighbour* and a *Support Vector Machine* classifier to predict the function of proteins using text features. In order to classify proteins that have multiple functions, we modified the traditional *k-Nearest Neighbour* algorithm to allow

multi-class classification. We also proposed a simple metric to assign scores to reflect the confidence of the predictions made by the *k-Nearest Neighbour* classifier. In-depth details about the classifiers are provided in Section 3.2.

3. We evaluated the performance of our system using a dataset that consists of proteins, along with their functional annotations from the manually curated protein database *UniProtKB/Swiss-Prot*. The resulting dataset contains 36,536 proteins that were used to test the accuracy of our classifier. To show the relative performance of our classifier, we compare it to two other baseline classifiers. The first baseline classifier assigns function at random based on the prior distribution of the function classes, and the second baseline classifier uses sequence similarity to predict function through homology based transfer.
4. We used our classifier to predict the function of proteins given by CAFA organizers and submitted the predictions as part of the CAFA 2011 competition. As a result of the competition, we were invited to present our function prediction system at the CAFA SIG at ISMB 2011 (Wong and Shatkay, 2011) as well as a journal publication (Wong and Shatkay, 2013).

1.3 Thesis Outline

The thesis is organized as follows: In Chapter 2 we provide background about proteins and discuss related work on protein function prediction. In Chapter 3 we describe the implementation of our system for protein function prediction. We then present the evaluation results of our system in Chapter 4. We conclude the thesis and discuss future work in Chapter 5.

Chapter 2

Background and Related Work

In this chapter, we provide background information about the basic biochemistry of proteins and present several of the methods for studying and annotating protein function. We start with a description of proteins in Section 2.1 and discuss the importance of studying their function in Section 2.2. In Section 2.3, various annotation schemes that are commonly used to describe protein function are presented. Next, in Section 2.4, we review common experimental assays for determining protein function and present the need of computational systems to assist in its prediction and annotation. In Section 2.5, we survey examples of different computational systems and the different types of features that are most broadly used to predict protein function

2.1 Biochemistry of proteins

Proteins are responsible for all vital functions within the living cell. They serve as messengers, provide means of energy storage, support the cell structure and carry out numerous other fundamental roles.

Proteins are sequences of amino acids. There are twenty different amino acid residues and each has unique chemical properties: some are hydrophobic while others form hydrogen bonds with each other; some are positively charged while others are negatively charged. Based on their chemical properties, the amino acids within a protein sequence interact with each other, causing the protein sequence to fold into a three-dimensional structure. Since each protein has a unique sequence of amino acids, its three-dimensional structure is unique as well. The unique structure of different proteins is the primary source of diversity in protein functions. Proteins with different structures bind to different molecules and accordingly perform different functions. Some

proteins act as enzymes that help break down molecules while other proteins bind with each other to form polymers that act as structural support. As a result of the relationship between sequence, structure, and function, protein sequence and structure data are often used to infer protein function (Solomon et al. 2002).

Genes are stretches of DNA found within an organism's genetic material that encodes for proteins. Recent advances in genome sequencing technology leads to rapid growth in the amount of sequenced genes, giving rise to many proteins whose sequence is known but whose function is yet to be determined.

2.2 Importance of Studying Protein Function

The role of proteins extends beyond regulating cellular processes, as proteins are directly involved in maintaining the health of an organism. In fact, many diseases can be traced back to dysfunctional proteins. For example, the *insulin* protein, produced in the pancreas, is a hormone that helps regulate sugar metabolism within the body. An *insulin* protein that is not functioning properly, or is missing, causes diabetes. Studying protein function is therefore important as it gives us a better understanding of how proteins affect health and disease (Jimenez-Sanchez et al., 2002; Ng and Henikoff, 2002).

Furthermore, understanding the function of proteins also assists in the process of designing new drugs and treatment options. For example, the discovery that the *tumor-necrosis-factor-alpha* (*TNF-alpha*) protein plays a functional role in inflammatory response eventually led to developing drugs that target *TNF-alpha* for treating arthritis (Feldmann and Maini, 2003). Another example involves the protein *erythropoietin*, which stimulates the production of red blood cells. This knowledge inspired the development of Epogen, a drug that mimics the function of *erythropoietin* and is used to treat anemic patients (Jelkmann, 2007).

Since proteins play such a significant role in human health, there is an ongoing effort to discover and annotate the function of sequenced proteins. Protein function can be determined using both experimental methods (Petsko and Ringe, 2004) and computational systems (Pellegrini, 2001). In this thesis, we focus on computational systems that aim to predict and annotate protein function.

2.3 Annotating Protein Function

Notably, the term "protein function" is not very-well defined. While the *chemical function* of a protein refers to the chemical reaction that the protein partakes in, the *biological function* is referring to the protein's role in biological processes. Furthermore, a single protein may have multiple functions. For instance, the protein *Tubulin* is an enzyme whose chemical function is the hydrolysis of *guanosine-5'-triphosphate* (GTP) molecules. However, in terms of its biological function, it is a structural protein that is involved in cellular transport, cell division, and cell mobility (Petsko and Ringe, 2004). To address the ambiguity involved in defining a protein's function, several standardized annotation schemes have been proposed. In this section, we describe three annotation schemes: Enzyme Commission Classification, Functional Catalogue and Gene Ontology.

The earliest annotation scheme of the three is the Enzyme Commission Classification (Webb, 1992). Under this annotation scheme, the enzymatic function of proteins is described using Enzyme Commission numbers. The Enzyme Commission numbers consist of four digits corresponding to four hierarchical categories. The first digit represents the most general category for describing the enzyme function. Each subsequent digit to the right denotes a more specific category.

An alternative scheme for annotating protein function is the FunCat, (Rueep et al., 2004). Unlike the Enzyme Commission Classification, which are used to classify proteins according to their enzymatic activity, FunCat categorizes protein function according to the biological processes that proteins are involved in (Ruepp et al., 2004). There are 28 main categories in FunCat, which include functions such as *transport, metabolism, developmental processes*, and *information pathways*. Each of the 28 categories is further divided into subcategories that are organized in a tree-like hierarchical structure. The higher levels of the hierarchy are general, whereas the subcategories at the lower levels are more specific. Similar to the Enzyme Commission system, each subcategory in FunCat is assigned a digit. For example, the sequence of digits *01* represents the highest level category, *metabolism*, whereas a much more specific description like *biosynthesis of glutamate* is represented as *01.01.03.02.01*.

The last annotation scheme that we review is the Gene Ontology (GO, Ashburner et al., 2000), which is the one most widely used. This is also the annotation scheme that we use to define function classes for our function prediction system. One of the advantage of GO is that it provides a standardized vocabulary of GO categories for annotating both the biological function and the molecular function of proteins. GO is organized as three sub-ontologies: *molecular function*, *biological process*, and *cellular location.* Even though *cellular location* is not strictly a "function", it is included in GO because protein function depends on cellular location. Each of the sub-ontologies is organized as a hierarchical, directed acyclic graph where each node in the graph corresponds to a different GO category. The category at the root node is more general, and the categories become increasingly specific further down the graph.

2.4 Experimental Methods for Determining Protein Function

There are two main categories of techniques for determining protein function: *experimental* and *computational*. *Experimental* techniques refer to experiments that are conducted in a laboratory setting to investigate protein function, while *computational* techniques refer to computer-based methods that algorithmically predict function using existing knowledge about the protein.

Traditionally, determining a protein's function experimentally involves experiments that are first performed to determine its fundamental characteristics including its amino acid sequence, structure, subcellular location, or molecular interaction. The information gathered from these experiments is then used to deduce the protein's potential molecular function. The hypothesized function of a protein is then confirmed using gene-knockout or gene-knockdown experiments. In these experiments, the expression level of a protein of interest is decreased or suppressed in order to observe the physiological effects (Alberts et al., 2002).

In gene-knockout experiments, the production of the protein of interest is halted by mutating its encoding gene (Carpenter & Sabatini, 2004). In gene-knockdown experiments, the number of produced proteins is drastically reduced by using techniques such as RNA interference (McManus and Sharp, 2002). However, knock-out and knock-down experiments are not always effective because these experiments may kill the organism or the cell before any physiological effects can be observed. Also, certain cell types are more resistant to the effects of RNA interference (Fraser et al. 2000).

Although experimental methods for determining protein function are the most accurate, they are time consuming and resource intensive. Given the number of sequenced proteins whose function is still unknown, experimental methods alone cannot uncover the function of all the

proteins in a timely manner. The need for high-throughput methods for studying protein function has driven the development of computational systems that predict protein function using existing knowledge. The predicted function can then be used to further guide investigation, and can be confirmed using experimental methods.

2.5 Homology-based Methods for Determining Protein Function

To try and elucidate the function of proteins, computational methods utilize known properties of proteins such as sequence, structure, interactions, and evidence from literature to predict function. In Chapter 1, we introduced two general categories of computational prediction systems: systems that assign protein function based on homology and systems that predict function using features that are derived from known information about proteins. In this section, we review homology-based methods.

The most common approach to computationally assign function to a protein is to transfer the functional annotation of homologous proteins to the yet unannotated protein (Loewenstein et al. 2009). Homologous proteins are proteins (in different species, or within the same species) that evolved from the same common ancestor. There are two types of homologs: orthologs and paralogs. Orthologous proteins are the result of speciation, and retain the same function in the different species, whereas paralogous proteins are the result of gene duplication and have different functions. The assumption underlying homology-transfer is that proteins with similar sequences are indeed orthologs that retain the same functions. Computer-based systems that use homologous proteins to assign function infer homology by measuring the similarity between protein sequences or structures. In the following section, we first discuss systems that use sequence similarity and then describe systems that use structural similarity.

2.5.1 Function Prediction using Sequence Similarity

Sequence similarity can be assessed using alignment programs such as BLAST (Altschul et al., 1990) to search the UniProtKB/SwissProt databases (Apweiler et al. 2004) for proteins with similar subsequences. The alignment programs return a set of proteins that have similar sequences along with metrics such as *expectation value* and *sequence identity percentage* to indicate the degree of sequence similarity between two proteins. Computational prediction systems often employ these metrics to measure the confidence of the functional annotation. Examples of function prediction systems that use sequence similarity include GOtcha (Martin et al. 2004), OntoBLAST (Zehetner, 2003), and BLAST2GO (Conesa et al. 2005).

GOtcha is one of the most commonly used prediction system that assigns function annotations using sequence similarity. GOtcha assigns functional annotation to proteins using GO categories from all three sub-ontologies of GO. Given an uncharacterized protein *p*, GOtcha uses BLAST to find annotated proteins that had similar sequences to *p*. The GO categories that are associated with each matched protein are assigned a score proportional to the degree of similarity between sequences. The score for each GO category across all matched proteins are summed together and used to output a ranked list of GO categories for protein *p*. The performance of the system was evaluated using about 39,000 annotated proteins from seven different organisms. At a cut-off score of 50%, GOthca had a Recall of 0.47 and a Precision of 0.61. The evaluation metrics *Recall* and *Precision* are defined later in Section 4.1.

All other computational systems that annotate function based on sequence similarity use the same fundamental principles as GOtcha. They use BLAST to find annotated proteins that are similar to the target protein of interest and the GO categories of the annotated proteins are scored based on sequence similarity and assigned to the target protein. The systems differ from one

another in the specific ways in which the GO categories are scored and assigned. For example, in the system OntoBLAST, the *expectation values* of the sequence alignment is used to determine the score for each GO category, whereas in BLAST2GO, the GO categories are scored using the highest *sequence identity* of all aligned sequences.

The problem with using sequence similarity to predict protein function is that, the assumption that all homologous proteins are orthologs that retain the same function does not always hold true. In reality, proteins that have similar sequences may be paralogs that have different functions. Moreover, Rost (2002) showed that proteins with high sequence similarity may have different functions. A potential explanation for this phenomenon is that only a small portion of the protein sequence is responsible for the functional structure of the protein (Friedberg, 2006).

2.5.2 Function Prediction using Structure Similarity

An alternative homology-based method is to use structural similarity to predict function. The advantage of using structural similarity is that structures tend to be more highly conserved than sequences among proteins with the same function. In fact, proteins whose sequences are different can still have similar structures and functions (Brenner et al. 1996). If structural information is available for a protein whose function is unknown, structural alignment programs such as FATCAT (Ye and Godzik, 2004) or CATHEDRAL (Redfern et al., 2007), can be used to find proteins with similar structure. In order to calculate structural similarity, protein structures are first represented using distance matrices. These matrices are 2D representations of protein structures, containing for each pair of amino acids in the protein the distance between their respective alpha carbon. (The alpha carbon is the carbon to which the functional group of the

amino acid is attached). The distance matrix representations are compared in order to measure similarity among the respective proteins.

Even though there are many systems that identify structural similarities among proteins (Friedberg, 2006; Gherardini and Helmer-Citterich, 2008), AnnoLite (Marti-Renom et al. 2007) is the only system that uses structural similarity to annotate protein function. AnnoLite uses the DBAli database (Marti-Renom et al. 2001), which is a database of protein structure alignments, to find proteins that have similar structure to an uncharacterized protein. The proteins are then ranked according to their *sequence identity* to the query protein. The GO categories that are most significantly associated with the 25 highest ranked proteins are assigned to the query protein. The performance of AnnoLite was tested using four different datasets, two with about 2000 proteins each, and two with about 4000 proteins each. The Precision of AnnoLite in predicting GO categories is 0.75 with a Recall of 0.85.

While predictions based on structural similarity can be more accurate than predictions made using sequence similarity, the downside is that structural data is often not available.

2.6 Machine-Learning Based Methods for Predicting Protein Function

In the absence of homologous sequence and structure, homology-based function prediction systems cannot be used. For such newly discovered proteins, function prediction systems that use machine-learning classifiers, as described below, are more suitable. Machine-learning based prediction systems attempt to use protein features that can discriminate among proteins with different functions as a basis for assigning a function to a yet-unannotated protein. The protein features can be derived from different sources such as the amino-acid sequence, protein structure or interaction data, and text associated with the protein.

2.6.1 Function Prediction using Sequence-Derived Features

Sequence-derived features include: amino acid composition, sequence length and protein sorting signals. The assumption behind using sequence-derived features to determine function is that if two proteins share a similar function, they will have similar signals in their sequences.

This strategy was first employed by Jensen et al. (2002) in the system ProtFun. ProtFun uses an ensemble of five *Neural Network* classifiers to assign proteins into one of 14 GO function classes. In addition, if a protein is predicted to be an enzyme, its enzymatic function is assigned an Enzyme Commission number as well. ProtFun uses 14 sequence-derived features in total. The sequence-derived features include the number of positively and negatively charged residues, hydrophobicity of the sequence, predicted regions of low complexity, predicted post-translational modification sites, and predicted secondary structure. The performance of ProtFun was evaluated using a test set of about 5500 human proteins from UnitProtKB/Swiss-Prot (Apweiler et al. 2004). At a Recall of 0.80, the false positive rate for the different function classes ranged from 0.10 to 0.70, with an average false positive rate of about 0.40.

Another prediction system that uses sequence-derived features is FFPred (Lobley et al, 2008). FFPred covers a wider variety of function classes than ProtFun using 197 function classes. Of the 197 classes, 111 are GO *molecular function* terms and 86 are GO *biological process* terms. For classification, FFPred uses *Support Vector Machines* (Webb et al., 2011) instead of the *Neural Networks* classifier (Bishop, 2006) used by ProtFun. For each GO category, five different SVMs are trained to separate the proteins that belong to the GO category from the proteins that belong to other GO categories. If a majority of the five SVMs classify the query protein as a positive example, then the GO category is assigned to the query protein. The performance of

FFPred was evaluated using a dataset of 102,173 annotated proteins from the Gene Ontology Annotation database (Camon et al. 2004), which included proteins from six different organisms. The predictions made on human proteins had the highest Recall and Precision out of the six organisms with an average Recall of 0.67 and an average Precision of 0.68 across all GO terms classified.

2.6.2 Function Prediction using Structure-derived Features

In this section, we review two different systems, PHUNCTIONER (Pazos and Sternberg, 2004) and ConFunc (Wass and Sternberg, 2008). Both systems use structural domains that are associated with protein functions as features to predict protein function. The motivation behind these methods is that proteins with similar functions bind to the same type of molecules and tend to have similar structural domains.

In order for PHUNCTIONER to predict function, structural domains that are associated with annotated proteins were first extracted from the Families of Structurally Similar Proteins database (Holm *et al.*, 1992). This resulted in a dataset of 4753 structural domains spanning 121 unique GO annotations. The collection of structural domains was then used to assign function annotations. Given a protein whose function is not yet known, GO categories are assigned to the protein based on the structural domains that are present. The performance of PHUNCTIONER was evaluated using test sets ranging from 2011 to 6168 proteins. At a score threshold of 6.0, 88% of the predicted GO categories were true positives at a Recall of 0.57.

An alternative approach was used in ConFunc, where multiple sequence alignments were used to extract structural domains. First, the authors compiled a dataset of annotated proteins and split them into subsets according to their GO functions. They then performed multiple sequence alignment on each subset to find the structural domains that are associated with each GO

annotation. Given an uncharacterized protein, ConFunc assigns GO annotations to the proteins using a similar strategy to the one used by PHUNCTIONER. The performance of ConFunc was tested on a set of 7150 protein sequences. Recall ranged from 0.37 to 0.54 while Precision ranged from 0.34 to 0.70. The highest Precision for ConFunc was 0.70 at a Recall of 0.30.

2.6.3 Function Prediction using Protein-Protein Interaction Features

Aside from protein sequence and structure, protein-protein interaction data obtained from experimental techniques are also used as features. Protein-protein interaction data show which proteins physically interact with one another. If two proteins interact, they are assumed to be part of the same biological pathway and, therefore have a similar function. Based on this assumption, computational systems use protein-protein interaction data to transfer the function annotations of proteins to interacting proteins whose function are yet unknown.

Schwikowski et al. (2000) were the first group to adopt this strategy in their prediction system. Using available protein-protein interaction data, their system first creates an interaction network and then uses it to transfer function annotations. To do so, the functions of proteins that are direct neighbours of an uncharacterized protein are tallied and the three most common functions are assigned to the uncharacterized protein. For their evaluation, they used a unique annotation scheme that they developed and consequently, it was difficult to compare the performance of their system to other function prediction systems.

In 2006, Chua et al. improved upon this strategy by incorporating the annotated functions of *level-2 neighbours* of uncharacterized proteins (indirect neighbours that are two interactions apart and interact with a common protein) into the function prediction process. Their rationale for using *level-2 neighbours* is that since these neighbours and the uncharacterized protein interact with the same common protein, then it is likely that they have similar biochemical function. In

their system, the functions of direct neighbours and level-2 neighbours are both assigned to the uncharacterized protein. The performance of the system was evaluated on a dataset of 4162 proteins taken from the Comprehensive Yeast Genome Database of the Munich Information Center for Protein Sequences (Mewes et al., 2010) using 117 FunCat categories. Their results showed that incorporating *level-2 neighbours* improved the Precision to around 0.30 at a Recall of 0.50 compared to a Precision of around 0.10 at the same Recall when only direct neighbours are used.

2.6.4 Function Prediction using Heterogeneous Sources of Features

As each type of protein features has its own advantages and disadvantages, some function prediction systems combined different types of features together. For example, Guan et al. (2008) used a combination of features including sequence, structure domain, and protein-protein interaction data as well as phenotype and phylogenetic data to represent proteins. For function classes, they used a total of 1726 *biological process* categories, 326 *cellular component* categories, and 763 *molecular function* categories from GO. For each GO category, a SVM classifier was trained using proteins that were annotated with either the GO category itself or one of its children as positive examples. Proteins that were annotated with any other GO category were used as negative examples. Their system was evaluated using a set of 1954 proteins. The average Precision across all the GO categories was 0.13 when measured at a Recall of 0.20. Even though the average Precision seems rather low, their system was among the top three among nine other methods that participated in MouseFunc project (Pena-Castillo et al., 2008).

Other groups that use heterogeneous sources of data as features include Pavlidis et al. (2001), who combined microarray expression data and phylogenetic data and used SVM classifiers to predict function, as well as Yao and Ruzzo (2006), who represented proteins using

protein sequence and microarray expression data and made predictions using *k-Nearest Neighbour* classifiers (Webb et al, 2011).

Even though the different groups used different combinations of data as features, a common observation shared among the different groups is that combining heterogeneous sources of data to predict function improves prediction performance.

2.6.5 Function prediction using text data as features

The last type of systems that we discuss here are those that use features derived from text sources, such as protein databases and biomedical literature, to predict protein function. Systems that use text features can be separated into two sub-categories: systems that perform *information extraction* and systems that perform *classification* using machine-learning techniques.

The goal of *information extraction* systems is to recognize and retrieve text from database or biomedical literature that is related to protein function. This includes systems that use natural language processing techniques to match and identify sentences that discuss protein function as well as systems that use dictionary-based techniques to associate keywords extracted from databases and abstracts with protein function.

One of the earliest information extraction systems, AbXtract (Andrade and Valencia, 1998) identifies sentences from biomedical abstracts that discussed the function of proteins and returns a ranked list of sentences ranked according to the statistical significance of the words that were present. Since then, different groups experimented with alternative strategies to retrieve relevant passages of text. For example, Chiang et al. (2003) used a pattern-matching algorithm to identify sentences that contain both a protein name and a GO category. Another group, Koike et

al. (2005), used rule-based techniques to analyze sentence structure and extracted sentences that mentioned an association between a protein and a GO category.

There are also systems that extract keywords from literature or databases in order to associate the keywords with protein function. For example, the system developed by Perez et al. (2004) created a dictionary by mapping the MeSH keywords associated with PubMed to GO categories. Using the dictionary, their system assigns GO categories to proteins when MeSH keywords are available for the proteins' associated abstracts. They evaluated their system using 6042 proteins that were annotated with at least one GO category and reported a *Precision* of 0.68 but the *Recall* is only 0.08.

Alternatively, the system by Groth et al. (2008) assigns function annotations by clustering genes that have similar textual descriptions together and transferring the functions of annotated genes to the uncharacterized genes that belong in the same cluster. Their evaluation was performed on a dataset of 4,438 genes that were clustered into 295 groups, where 90% of the genes were used for training and 10% were used for testing. This was repeated 200 times using different training and test sets and the average Precision and Recall were computed. Of the 295 groups, no significant predictions were made for 99 of them. For the remaining 196 groups, the Precision was around 0.58 and the Recall was around 0.23. Many other types of information extraction systems are used within the biomedical domain to discover biological knowledge (see for instance, surveys by Jensen *et al.* (2006) and by Cohen *et al.* (2005)).

As for systems that perform classification using text features, several systems already exist. However, these systems were limited in scope as they were only used to annotate yeast proteins and were evaluated on a small number of proteins. For example, Raychaudhuri *et al.* (2002) developed a *Maximum Entropy* classifier that assigns *biological process* GO categories to

abstracts. The GO categories assigned to the abstracts were also transferred to the proteins that are referenced in the abstracts. Their system used 21 *biological process* GO categories as function categories and was only tested on 1188 *yeast* proteins. Their results showed that the prediction accuracy varies greatly with the number of abstracts that were available for training; their best prediction accuracy was 72.12%.

Similarly, Izumitani et al. (2004) created a text-based prediction system that classifies yeast genes into GO categories using *Support Vector Machine*. In their system, the genes were represented using words that were extracted from associated abstracts. In order to classify genes into GO categories, a SVM classifier was trained for each distinct GO category. The performances of 21 GO categories were evaluated using 3,295 genes. The computed results showed *Precision* that ranged from 0.59 to 0.72 and a *Recall* that ranged from 0.56 to 0.73.

The last prediction system that we review in this section is EpiLoc (Brady and Shatkay, 2008), which is strongly related to the system that we developed for this thesis. EpiLoc is a text-based classification system used to predict protein subcellular localization. In this work, protein data was collected from the curated database UniProtKB/Swiss-Prot (Apweiler et al. 2004), and was used to train a SVM classifier. In order to represent the proteins using text, the abstracts that are referenced from each protein entry in the UniProtKB/Swiss-Prot database were retrieved from PubMed. The terms from the associated abstracts that were statistically significant were then selected as text features using the *Z-Score* statistical test. While the focus of EpiLoc was protein subcellular localization as opposed to function, they showed that representing proteins using text features offered competitive performance when compared to other subcellular localization prediction systems that use other types of protein features.

To this end, given the limited number of text-based prediction systems that predict the function for proteins from different organisms, we developed a system that assigns GO categories to proteins from *any* organism using text features. The implementation of our system is explained in detail in the following chapter.

Chapter 3

Methods

This chapter describes the implementation of the text-based protein function prediction system we developed for this thesis. In Section 3.1, the method for representing proteins using features derived from text is described in detail. We begin by describing how the dataset of proteins and their associated text are collected. Then, we explain how the function classes of proteins are defined according to their GO annotations. Next, the feature selection process is presented in detail as we show how distinguishing terms were extracted from associated text and how we used the terms to represent proteins. Last, we describe how proteins that lack related text are handled.

In Section 3.2, we introduce the *Support Vector Machine* and *k-Nearest Neighbour* classifiers that we used for our prediction system. We elaborate on the modifications that we performed to the *k-Nearest Neighbour* classifier to enable multi-class classification and explain how confidence scores were calculated for each prediction.

3.1 Representing Proteins using Text Features

In order to train a text-based classifier to predict protein function, we compiled a dataset of proteins with known functions. The dataset of proteins was used to train and test our classifier using five-fold cross-validation, as described later in Section 4.1. For each protein in the dataset, we retrieved the abstracts of associated published articles and selected from the abstracts *distinguishing terms* that best characterize proteins with different functions. The feature selection process is described in further detail below.

3.1.1 Compiling the dataset of proteins

To build our dataset of proteins, we searched UniProtKB/Swiss-Prot (Apweiler et al. 2004) for a list of candidate proteins whose function is known, that is, proteins already annotated with at least one eligible GO category. Additionally, the candidate proteins must also have at least one reference to an article in the literature listed in its UniProtKB/Swiss-Prot entry. As previously mentioned, GO is an ontology of terms for describing protein function, and is organized as three sub-ontologies: *biological process, molecular function*, and *cellular component*. In this thesis, we are only interested in proteins that are annotated with GO categories from the *biological process* and *molecular function* sub-ontologies.

Furthermore, to ensure that the dataset consists of proteins whose annotation is of high certainty, we checked the evidence code associated with each GO annotation. Evidence codes are part of every GO annotation and are manually assigned by curators to indicate the type of evidence available that supports the annotated function. For our dataset, we only included proteins with GO annotations that were inferred from biological experiments, inferred by curators from indirect evidence that is available (i.e. experimental evidence or literature that support the function of a protein belonging to the same pathway), or supported by traceable statements from authors. We excluded protein with GO annotations that were generated by computational method or supported only by non-traceable author statements. The evidence codes that were included in the dataset are shown in the left column of Table 1, while the evidence codes that were excluded from the dataset are listed in the right column.

Reliable evidence codes		Unreliable evidence codes	
EXP	Inferred from Experiment	ISS	Inferred from Sequence/Structural Similarity
IDA	Inferred from Direct Assay	ISO	Inferred from Sequence Orthology
IPI	Inferred from Physical Interaction	ISA	Inferred from Sequence Alignment
IMP	Inferred from Mutant Phenotype	ISM	Inferred from Sequence Model
IGI	Inferred from Genetic Interaction	IGC	Inferred from Genomic Context
IEP	Inferred from Expression Pattern	RCA	Reviewed Computational Analysis
IC	Inferred by Curator	IEA	Inferred from Electronic Annotation
TAS	Traceable Author Statement	NAS	Non-traceable Author Statement

Table 3.1.1: The left column shows GO evidence codes that are considered reliable and included in our dataset. The right column shows evidence codes that are considered unreliable and are excluded.

Since our goal was to identify from each protein in the dataset text features that can best characterize proteins from a single GO category, we excluded from our dataset proteins that had three or more annotations from the same GO sub-ontology. After removing proteins that did not meet the listed criteria, a total of 36,536 proteins remained as our final dataset. In our final dataset, there are 21,764 proteins with *biological process* categories and 22,309 proteins with *molecular function* categories.

3.1.2 Retrieving associated text for our dataset of proteins

Next, we compiled a corpus of biomedical abstracts from which text features can be extracted to represent the proteins. We used abstracts rather than full–text articles because the abstracts are readily available in public databases, whereas full-text articles are often not freely available. The use of abstracts as a text source has shown to be beneficial in an earlier work on protein subcellular location prediction (Brady & Shatkay, 2008). Moreover, Shah *et al.* (2003) showed that the abstract has the highest density of keywords out of all other sections in an article and therefore is typically a good source of text features. To retrieve the associated abstracts, we

first extracted the PubMed identifiers, provided as references into the literature, from the UniProtKB/Swiss-Prot entries for each protein. We then retrieved the corresponding abstracts that were all publicly available from PubMed. However, not all of the retrieved abstracts were useful for training because some of them are associated with multiple functions. As previously mentioned, our aim was to extract *distinguishing terms* from the associated abstracts that were predictive of different GO categories. Therefore, we removed abstracts that were associated with more than three GO categories. In total, our final corpus consists of 68,337 abstracts for all the proteins.

3.1.3 Defining Classes for Protein Functions

For our function prediction system, our function classes consist of GO categories from the *biological process* and *molecular function* sub-ontologies. In contrast to Brady and Shatkay's (2008) text-based protein localization prediction system that focused on a relatively small number of organelles as classes (at most 13), there are about 20,000 distinct *biological process* and 9000 *molecular function* GO categories. Ideally, our goal was to use each of the individual GO categories found in the *biological process* and *molecular function* sub-ontologies as a separate function class. However, many of the specific GO categories do not have a sufficient number of proteins (and associated abstracts) that can be used to train our classifier. For example, for the GO term '*platelet activating factor metabolism',* there is only one associated protein. The low number of associated proteins prevented us from extracting statistically significant text features using our feature selection method.

For this reason, we used a smaller subset of GO categories as our function classes. We explored different strategies for reducing the total number of function classes. First, we tried selecting as our function classes the top 20 GO categories that have the highest number of

associated proteins. However, we noticed that the selected GO categories only represented a small subset of the protein functions found at a higher level of the Gene Ontology. Specifically, when we traced the selected GO categories to their parents at the second level of the GO hierarchy, we found that only three out of the 17 categories (ex. '*binding*', '*structural molecular activity'*, and '*electron carrier activity*') from the *molecular function* sub-ontology were represented. For the *biological process* sub-ontology, the selected categories only spanned five out of the 29 categories at the second level of the GO hierarchy. Thus we did not choose this strategy to define our function classes, as it would greatly limit the diversity of protein functions our classifier can assign.

To ensure that the diversity of protein functions is better accounted for, we tried using all the GO categories at a specific level of the GO hierarchy as function classes. Initially, we tried using the third level of the GO hierarchy but we found that even at the third level, a majority of the classes did not have a sufficient number of associated proteins (fewer than 10 proteins for most of the categories) available for training. We thus used as function classes only the GO categories at the second level of the GO hierarchy (one level away from the root node), merging together all the descendant GO categories below each node all the way down to the leaf-nodes.

At the second level of the GO hierarchy, there are 29 distinct *biological process* categories and 17 distinct *molecular function* categories. However, 12 of these categories each had fewer than 15 proteins and were therefore removed (along with the 52 proteins that were associated with them) from our dataset. This left us with a final total of 24 *biological process* categories and 10 *molecular function* categories.

3.1.4 Selecting Text Features from Abstracts

After the function classes were defined, we extracted from the associated abstracts text features that can represent proteins of different function classes. Rather than use all the terms from the abstracts as features, we removed, through preprocessing, terms that have little power in distinguishing between proteins with different functions. Then, we applied feature selection to identify *distinguishing terms* that best characterize the different function classes. In order to represent proteins using the distinguishing terms, we used the '*bag of words*' approach (Lewis 1998; Mitchell, 1997). In other words, proteins were represented as a vector of term weights where each term weight corresponds to the frequency of the distinguishing term within the protein's associated abstracts.

Text Preprocessing: First, we parsed our abstracts into individual words (unigrams) as well as pairs of consecutive words (bigrams). We then removed stop words such as pronouns, prepositions, and numbers. The complete list of stop words that were removed is available in *Appendix A*. We also removed common words that were found in more than 60% of the abstracts and rare words that were found in fewer than three abstracts. The rare words were removed because they are not useful in representing the majority of the proteins. The common words and stop words were removed because they are not very effective for distinguishing between different function classes.

Next, we applied Porter Stemming (Porter, 2006), which stripped off suffixes so that words with different suffixes but the same semantic meaning can be represented as a single term using its root. For example, the words 'transporting' and 'transported' were both reduced to 'transport' after Porter Stemming.

Text Feature Selection: After preprocessing the text, we applied feature selection to select a set of terms that can best distinguish between proteins from different function classes. Since the *biological process* categories and the *molecular function* categories describe different aspects of protein functions, feature selection was performed separately for the two sub-ontologies. The advantage of applying feature selection is that it removes terms that have little distinguishing power and keeps only the terms that characterize the different protein functions well. Applying feature selection was shown to improve the overall accuracy of classifiers (Yang & Pedersen 1997).

There are many different methods that can be used for feature selection and each produces a different set of text features and affects the classifier's performance in different ways. In Brady's thesis (2007), he compared the effectiveness of several different statistical methods for feature selection: *odds ratio, Chi-squared test, mutual information, information gain, Entropy* and *Z-score.* His results showed that the *Z-score, information gain,* and *Chi-squared test* methods are the top-performing methods. Therefore, for our function prediction system, we implemented the *Z-score* method (Devore, 2011) to select *distinguishing terms*. A term is considered *distinguishing* if the probability of the term to occur in the abstracts associated with a protein function *f*, is significantly different from its probability to occur in abstracts associated with all the other protein functions. Using the *Z-score* statistical test, we measured the significance of the difference between the probability of a term *t* to appear in an abstract associated with protein function *f*, denoted as Pr ($t \mid f$) and the probability of a term to appear in an abstract associated with another protein function *f ',* denoted as Pr($t \mid f'$).

The conditional probabilities Pr($t|f$) and Pr($t|f'$) were estimated using maximum likelihood estimates. The maximum likelihood estimate was obtained by dividing the number of

abstracts associated with function *f* and containing the term *t* by the total number of abstracts associated with function *f*. Formally, this is defined as:

$$\Pr(t \mid f) \approx \frac{|\, d \in D_f \quad s.t. \quad t \in d|}{|\, D_f \,|}, \tag{1}$$

where *d* denotes an individual abstract, and D_f denotes the set of abstracts that are associated with protein function *f*.

As previously mentioned, we used the *Z-score* test to determine the significance of the difference between the probabilities of the term to occur in abstracts associated with different functional classes. The *Z*-score is defined as:

$$Z^t_{f,f'} = \frac{\Pr(t \mid f) - \Pr\ (t \mid f')}{\sqrt{P \cdot (1-P) \cdot (\frac{1}{|D_f|} + \frac{1}{|D_{f'}|})}}, where\ P = \frac{|D_f| \cdot \Pr(t \mid f)\ |D_{f'}| \cdot \Pr\ (t \mid f')}{|D_f| + |D_{f'}|}. \tag{2}$$

A higher absolute value of the Z-score indicates a higher confidence that the difference between probabilities is statistically significant. Thus, for a term *t* appearing in abstracts associated with function *f*, if the absolute value of its Z-score is higher than a predetermined threshold for all other functions *f'*, *t* is selected as a distinguishing term for function *f*. When the threshold is set too low, too many uninformative terms are selected as distinguishing terms; when the threshold is set too high, some of the function classes may not have any distinguishing terms associated with them. To represent the proteins in our dataset, the union of all the distinguishing terms for all function classes was used as text features. For the *molecular function* classes, a total of 521 distinguishing terms were selected and for the *biological process* classes, a total of 831 distinguishing terms were used.

3.1.5 Representing Proteins using Text Features

The set of distinguishing terms, denoted as T_N, is used to represent each individual protein *p* using an *N*-dimensional vector of term weights (where *N = 521* for the *molecular function* classifier and *N = 831* for *biological process* classifier). Each term weight, $w_{t_i}^p$, represents the significance of term t_i, relative to the other distinguishing terms within the set of abstracts associated with the protein *p* (the set of abstracts is denoted as D_p). The term weight is calculated as the ratio between the number of times term *t* appears within D_p and the total number of all distinguishing terms, t_j, from the set T_N that appear in D_p. The term weight is formally defined as:

$$w_{t_i}^p = \frac{\#\ of\ times\ t_i\ appears\ in\ D_p}{\sum_{t_j \in T_N}(\#\ of\ times\ t_j appears\ in\ D_p)} \quad . \tag{3}$$

3.1.6 Proteins without associated text

While the proteins in our training set all have text associated with them, a query protein may not have any associated text for one of the following reasons:

- The protein is not found in the UniProtKB/Swiss-Prot database;
- There are no related articles recorded in the protein entry of the UniProtKB/Swiss-Prot ;
- The article(s) were removed from the text corpus as described in Section 3.1.2.

For an unclassified protein that has no associated text, to which we refer as *textless* protein, we adopted the same strategy used by Shatkay et al. (2008) and Brady (2007), which assigned the text features of a homolog protein to the *textless* protein. To determine the degree of homology between two proteins, we compare the amino acid sequences of the two proteins using BLAST. The assumption is that the greater the sequence similarity, the greater the likelihood that the two proteins are homologous.

We conducted BLAST search using BLASTP to compare the amino acid sequences of our *textless* protein to the sequences of our dataset of proteins that had associated abstracts. This returned a list of proteins with similar sequences along with an *expectation value* (*e-value*). The *e-value* indicates the expected number of times this match or a better one occurs by chance. In other words, a lower e-value means that the likelihood of the match to be a true homolog is greater. In addition to the e-value, we also considered the *sequence identity* to determine if the matched protein is homologous to the *textless* protein. The *sequence identity* measures the percentage of matched amino acids that is identical. A previous study by Brenner et al. (1996) suggested that the *sequence identity* of the match should be greater than 40% for a match to be considered homologous. Taking into consideration both the *e-value* and the *sequence identity*, we accepted a matched protein as a potential homolog only if the *e-value* was below 10 and the sequence identity was greater than 40%.

From the set of potential homologs, we took the three proteins with the lowest e-value and assigned a weighted combination of their feature vectors, v_i to the textless protein. Like Brady (2008), to account for the degree of homology between the textless protein and its homologs, we multiplied the term weights of each feature vector by the sequence identity, s_{v_i}. We then summed the resulting weighted feature vectors together and divided the term weights of each feature vector by three to obtain a vector representation for the textless protein $v_{textless}$. This is defined as:

$$v_{textless} = \frac{\sum_{i=1}^{3} s_{v_i} * (\, v_i)}{3} \,. \tag{4}$$

3.2 Machine-learning Classifiers

In this section, we describe the *k-Nearest Neighbour* (kNN) classifier and the *Support Vector Machine* (SVM) classifier that we used for our function prediction system. We chose to use kNN as well as the SVM classifier for our system because kNN is simple to use and it was shown to have comparable performance to SVM (Yao and Ruzzo, 2006).

3.2.1 Support Vector Machine

Support Vector Machines (SVMs) classifiers are obtained using supervised learning. Supervised learning requires a training dataset where the class of each example is known. For SVMs, the training dataset is used to construct a *hyperplane* that can separate the vector representation of data into two classes. The two classes can be any pair of classes from the training dataset, or *positives* and *negative* classes. A more in-depth description of class definitions is provided later in this section. In order to construct the hyperplane, the vectors are often mapped into a higher dimensional space where the vectors can be separated, using a *kernel function*. As there are many hyperplane that can separate the vectors into two classes, the hyperplane with the maximum margin is used by the SVM. The *margin* refers to the distance between the hyperplane and the nearest vectors of each class. For classification, a vector is assigned a class based on the side of the hyperplane on which it is located. (Figure 3.2.1)

As mentioned above, the hyperplane that is created during supervised learning can only separate the data into two classes. For classification problems that involve more than two classes, multiple SVMs are required. There are two main strategies for handling classification problems with multiple classes: *one-versus-all* and *one-versus-one*. For the *one-versus-one* approach, an SVM is trained for every pair of classes. In other words, for problems that have *n* classes, $\frac{n\,(n-1)}{2}$ SVMs are required. In order to classify a vector whose class is unknown, the results of all the

SVMs are combined using a voting strategy. The vector is presented to all the SVMs as input and the class assignment from each SVM is counted as a vote for the corresponding class. The class that has the most votes is assigned to the vector.

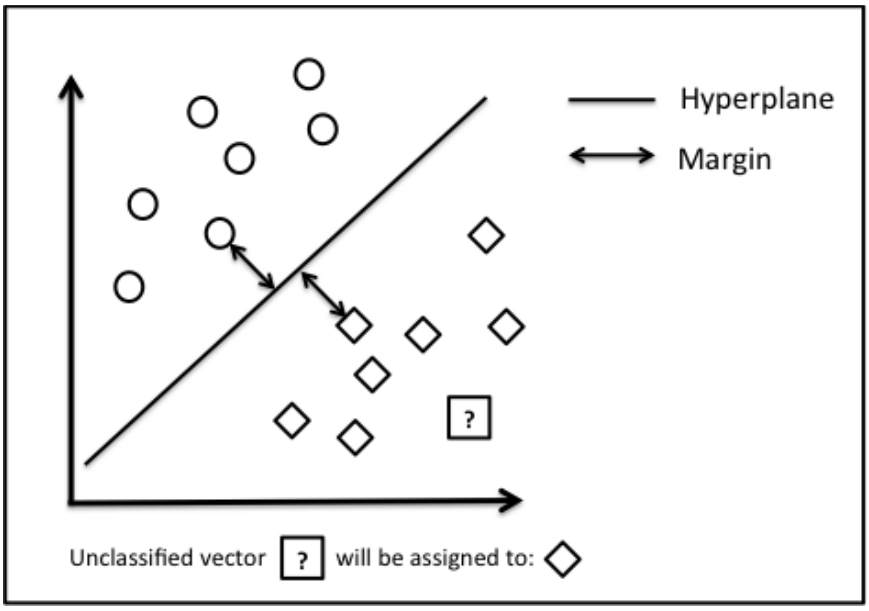

Figure 3.2.1: The data from two classes are separated by a hyperplane. The unclassified vector is assigned to a class based on its location relative to the hyperplane.

For the *one-versus-all* approach, a single binary SVM is trained for each class, to differentiate between vectors that belong to the class (positive class), and all the other vectors that do not belong to the class (negative class). This results in *n* SVM classifiers for problems that have *n* classes. Under this strategy, the SVM with the highest margin between the yet-unclassified vector and the hyperplane assigns the final class.

For our protein function prediction system, we used the LIBSVM (Chang and Lin, 2011) implementation of *Support Vector Machines*. LIBSVM uses the *one-versus-one* approach to handle multiple classes and it also provides an estimated probability of a vector belonging to a

class. For our *kernel*, we chose the *Radial Basis Function* (RBF), as it is often the default choice for SVM.

There are two parameters to optimize for the LIBSVM's implementation of Support Vector Machine, C and γ. The C parameter represents the cost function of the SVM and controls the tolerance for errors in the training data and the width of the margin. A higher value for C increases the penalty for misclassification in the training data and results in a narrower margin while a smaller value of C allows for more misclassification errors and results in a wider margin. The γ parameter relates to the RBF *kernel* and controls the width of the *kernel* function. A higher value of γ increases the likelihood of the SVM to over-fit the data while a smaller value of γ may cause the SVM to under-fit the data. Over-fitting occurs when the hyperplane of the classifier is too closely fitted to the features of the training data to the point that it does not generalize well for test data. Under-fitting occurs when the hyperplane does not separate the training data well and therefore has low prediction accuracy for both training and test data. For our text-based SVM classifier, we experimented with different values for C and γ and found that setting C to 0.8 and γ to 0.25 yielded the best classification performance.

3.2.2 The basic k-Nearest Neighbor approach

The *k-Nearest Neighbour* (kNN) classifier is another instance of supervised learning. In contrast to SVM, kNN classifier is a model-free method in the sense that it does not build a classification model from the training data. Therefore, there is no distinct training phase where the classifier 'learns' the parameters of a classification model. Instead, classification is performed ad-hoc during the test phase by determining the distance between the vector representation of the

unclassified protein and the vectors representing the training proteins, and finding the nearest neighbours. The unclassified protein is then assigned a function class according to the majority of function classes shared by its nearest neighbours.

The only parameters associated with a basic kNN classifier are the number of nearest neighbours to be considered, *k*. For our text-based kNN classifier, we experimented using 3, 5, 10, 15, and 20 neighbours and found that setting k to 10 yielded the highest accuracy.. For our implementation, we used the *knnclassify* package provided by *Matlab R2009b* (The MathWorks Inc, 2009) and we compared two of the most commonly used measures for determining nearest neighbours for kNN classification: *Euclidean* distance and *cosine coefficient* similarity. The *Euclidean* distance measures the planar distance between two *n*-dimensional feature vectors, $p = (p_1, p_2, \dots, p_{n-1}, p_n)$ and $q = (q_1, q_2, \dots, q_{n-1}, q_n)$, and is defined as:

$$\mathrm{euc}(p, q) = \sqrt{\sum_{i=1}^{n} \mid p_i - q_i \mid^2} \,. \tag{5}$$

The *cosine coefficient* is the cosine of the angle between two feature vectors. It is defined as:

$$\cos(p, q) = \frac{\sum_{i=1}^{n} p_i \times q_i}{\sqrt{\sum_{i=1}^{n} (p_i)^2} \times \sqrt{\sum_{i=1}^{n} (q_i)^2}} \,. \tag{6}$$

We found that for our classification task, both metrics yield similar results when used for finding the nearest neighbors, where the *cosine coefficient* similarity performs slightly better.

3.2.3 The modified kNN classifier

As described above, the basic implementation of the kNN classifier only assigns a single class to the query instance. This implementation is not suitable for protein function prediction because, as previously mentioned, a single protein can have multiple functions. Therefore, we implemented a simple modification to the kNN classifier to enable multi-class classification. The modified kNN classifier, with *k = 10*, finds for each yet-unclassified protein, represented as a

feature vector, its 10 nearest neighbors in the training set. If three or more of the 10 nearest neighbors belong to the same function class, then the function class is assigned to the unclassified protein. Under this classification scheme, a protein can be assigned a maximum of three distinct functions.

In addition to handling multi-class classification, we also modified the classifier to return a confidence score for each predicted class. For a protein, *p*, and a predicted function class, *f*, the confidence score $C_f(p)$ is formally defined as:

$$C_f(p) = 1 - \frac{\sum_{i=1}^{|N^f|} \cos\left(N_i^f, p\right)}{|N^f|} \ , \ 3 \leq |N^f| \leq 10 \, , \tag{7}$$

where out of the 10 nearest neighbours of the query protein *p*, $|N^f|$ is the number of nearest neighbors with function *f*, and $\cos\left(N_i^f, p\right)$ is the *cosine coefficient* between *p* and its *i*'th nearest neighbor with biological function *f*, N_i^f. The average of the *cosine coefficient* values between *p* and N_i^f is then used as the confidence score. For the work described throughout this thesis, we used a very simple formula to determine the confidence score as it was introduced primarily to meet the CAFA Challenge requirements. In the future, it will be worthwhile to experiment with different confidence scores. For example, one potential improvement is to redefine the formula such that the confidence score increases proportionately to the number of nearest neighbours that share the function *f*.

Chapter 4

Experiments and Results

In this chapter, we present the evaluation results for our text-based classifiers (*Text-kNN* and *Text-SVM*), and our baseline classifiers (*Base-Prior* and *Base-Seq)*. *Base-Prior* performs classification based on the prior distribution of the GO categories in our dataset, and *Base-Seq* assigns proteins into function classes based on sequence similarity. The baseline classifiers, along with the cross-validation dataset and performance metrics that we used are described in detail in Section 4.1. The performance metrics that we use to evaluate our systems are the standard *Precision*, *Recall*, *Matthew's Correlation Coefficient* (MCC), *F-measure* and overall accuracy.

In Section 4.2, we present the classification performance of *Text-kNN* on *molecular function* classes and on *biological process* classes using the cross-validation dataset. We compare its performance to the benchmark classifiers, *Base-Prior* and *Base-Seq.* When compared to the benchmark classifier *Base-Prior*, the text-based classifiers typically have a higher overall accuracy and average Precision but a lower average Recall. In comparison to *Base-Seq*, *Text-kNN* and *Text-SVM* have higher performance for *molecular function* classes but lower performance for a majority of the *biological process* classes. The text-based classifiers also perform better than the sequence-based classifier for small *biological process* classes.

In Section 4.3, we compare the performance of the two different classifiers we used for our text-based prediction systems, *Text-kNN* and *Text-SVM.* For *molecular function* classes, *Text-kNN* has a higher Recall than *Text-SVM* for the majority of the classes, but has significantly lower Recall, F-measure, and MCC. For *biological process* classes, *Text-kNN* does not perform as well as *Text-SVM* for most of the classes.

In Section 4.4, we show the evaluation results on proteins that lack associated text. .As explained in Section 3.1.6, proteins without any associated text are represented using the text features of homologous proteins (*textless)*. We compare the evaluation results of *Text-kNN* on textless proteins to the classification performance of *Text-kNN* on the cross-validation dataset. The evaluation suggests that the text-based classifier is equally effective at classifying textless proteins as classifying proteins with associated text into their respective function classes.

Last, in Section 4.5 we present the evaluation results from the CAFA Challenge in which we participated. At the beginning of the challenge period, CAFA organizers provided participants with a test dataset of more than 40,000 proteins that had no function annotations in UniProtKB. The participants used their function annotation system to predict the functions of proteins in the test dataset and submitted their predictions to the CAFA organizers. At the end of the competition, the CAFA organizers selected from the test dataset proteins whose functions were elucidated during the span of the competition and used them as the gold standard to evaluate the accuracy of the submitted predictions. In total, 596 proteins from the test dataset were annotated by the end of the challenge, of which 436 proteins were annotated with at least one *biological process* GO category, and 366 proteins were annotated with at least one *molecular function* GO category in the UniProtKB/Swiss-Prot database. After the evaluation was completed, the evaluation results were released to the participants. The CAFA evaluation results were consistent with our cross-validation results. The sequence-based classifiers typically demonstrated higher performance than our text-based classifier for most function classes. However, there were also a few function classes for which our text-based classifier showed significantly higher Precision and Recall.

4.1 Evaluating the Text-Based Classifiers

4.1.1 Benchmark classifiers

We compared the performance of our classifiers to two different baseline classifiers. The first baseline classifier assigns classes to a query protein according to the prior distribution of function classes in the dataset. For instance, if in the dataset 60% of the proteins belong to the function class '*binding'* and 40% of the proteins belong to the class '*catalytic activity'*, then the classifier assigns a class label to a protein *p* using Monte Carlo sampling from a label distribution with a 60% chance of obtaining the label '*binding*' and 40% chance of obtaining the label '*catalytic activity*'. The protein *p* is thus assigned the label '*binding*' by the classifier with a probability of 0.6 and the label '*catalytic activity'* with probability 0.4.

The second baseline classifier assigns function classes based on sequence similarity. Given an unclassified protein *p*, the classifier uses BLAST (with default parameters) to search for proteins with similar sequences to protein *p*. To ensure high quality of the alignments, the classifier excludes BLAST results that have *e-values* greater than ten or *sequence identity* lower than 40%. The classifier then transfers the protein function associated with the homologs to the yet-unclassified protein using the *k*-nearest neighbour classification scheme. The top ten proteins returned by BLAST are used as nearest neighbours (in order to give a fair comparison with our text-based kNN classifier), and the functions that are shared by at least three of the matched proteins are assigned to the unclassified protein *p*.

For the evaluation conducted by the CAFA Challenge, three different baseline classifiers were chosen by the CAFA organizers, denoted in this thesis as *CAFA-Prior, CAFA-Seq, and GOtcha. CAFA-Seq* and *GOtcha* are both sequence-based classifiers; they use BLAST to find proteins from UnitProtKB/Swiss-Prot whose sequences are similar to the target's, transferring the

functional annotations of the aligned proteins to the target protein. The main difference between the two classifiers is that *CAFA-Seq* uses the *percent identity* between sequences as a confidence score, whereas *GOtcha* bases its score on the sum of negative logs of e-values associated with the alignments between the target protein and the aligned annotated proteins. In contrast, *CAFA-Prior* assigns *every* GO category label to *each* protein in the dataset and uses the prior distribution of the GO categories in UnitProtKB/Swiss-Prot as a confidence score. A confidence score, associated with the function assigned by the classifier, aims to represent the confidence in the classifier's prediction. Typically, by requiring the classifiers to only report predictions whose confidence scores are above a minimum threshold the classifier's precision increases while its recall decreases. Thus, when comparing classifiers, a specified threshold is set such that only functions assigned with a confidence score higher than the threshold are evaluated. In the results discussed below, we compare classifiers' performance on the CAFA evaluation dataset using a confidence threshold of 0.95 for *molecular function* classes. For *biological process* classes, we use a lower confidence threshold for *Text-kNN*, *GOtcha*, and *CAFA-Prior* because no predictions were made at a confidence threshold of 0.95. The exact threshold used for each classifier is reported in Section 4.5.

4.1.2 Cross-Validation

We evaluated our classifiers and compared them to other baseline classifiers using stratified five-fold cross-validations. Stratified five-fold cross-validation means that the training dataset is partitioned into five disjoint subsets at random, where each subset retains the same distribution of class instances as in the original dataset. The classifiers' performance was evaluated five times; each time using a different subset as a test set, and using the remaining four subsets to train the classifier. Furthermore, to ensure that our results were not biased due to a

particular partition, we repeated the five-fold cross-validation five times using different five-way partitions. The performance from the different evaluations was then averaged to give a measure of the classifier's overall accuracy.

4.1.3 Evaluation Metrics

To assess the performance of the different classifiers at predicting each of the individual function classes we included the evaluation metrics *Precision*, *Recall*, *F-measure*, and *Matthew's Correlation Coefficient* (*MCC*) as was done by Brady (2007). In addition, we also used the F-measure as it is commonly used in conjunction with Precision and Recall (Forman, 2003).

All these measures are calculated based on the number of true positives (*TP*), false positives (*FP*), true negative (*TN*), and false negatives (*FN*). That is, for a given function class *f*, *TP* denotes the number of proteins whose function according to UniProtKB/Swiss-Prot is *f* and were labeled as *f* by the classifier; *FP* denotes the number of proteins that were labeled as *f* by the classifier but this label does not match their annotated function in UniProtKB/Swiss-Prot; *FN* denotes to the number of proteins whose function according to UniProtKB/Swiss-Prot is *f* but were mislabeled and assigned to another class by the classifier.

Precision denotes the fraction of the predicted proteins that actually belong to the function class. Recall denotes the fraction of proteins belonging to the function class that are correctly predicted by the classifier. The value of both Precision and Recall measures ranges from 0 to 1 where 1 corresponds to a perfect performance.

Precision is formally defined as:

$$Precision = \frac{TP}{TP+FP} \quad , \tag{7}$$

while Recall is formally defined as:

$$Recall = \frac{TP}{TP+FN} \quad . \tag{8}$$

The F-measure is the harmonic mean of Precision and Recall that combines the two evaluation metrics into a single measure. Formally it is defined as:

$$F = \frac{2P \cdot R}{P+R} \qquad . \tag{9}$$

Matthews Correlation Coefficient (MCC) (Matthews, 1975) is an evaluation measure that is often used to measure the performance of unbalanced datasets where the classes are of different sizes (Huang et al. 2010). The value ranges from -1 to 1 where a 0 corresponds to a completely random prediction. MCC is formally defined as:

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FN) \cdot (TP+FP) \cdot (TN+FN) \cdot (TN+FP)}} \quad . \tag{10}$$

We also measured the overall performance of the classifier using *overall accuracy*. For overall accuracy, we use the definition proposed by Rost (1993), O_{acc} = C / N, where C is the number of test proteins that are correctly classified and N is the total number of test proteins. The results from our evaluation are presented in the next Section.

4.2 Comparison between *Text-kNN* classifier and the baseline classifiers

In this Section, we compare the performance of *Text-kNN* to the baseline classifiers *Base-Prior* and *Base-Seq*. For *molecular function* classes, *Text-kNN* consistently outperforms *Base-Prior* according to all performance measures and have slightly higher Recall but lower Precision than *Base-Seq* on most classes. For *biological process* classes, *Text-kNN* also outperforms *Base-*

Prior on the majority of the classes. *Text-kNN* does not perform as well as *Base-Seq* for classes that have more than 200 associated proteins, but it does demonstrate a slightly better performance for classes that are smaller.

4.2.1 Performance of *Text-kNN* on *molecular function* classes

The overall accuracy of the classifiers *Text-kNN, Base-Prior,* and *Base-Seq* with respect to the *molecular function* categories is 62%, 43%, and 58% respectively. We note that the overall accuracy of an alternative *hypothetical Base-Prior* classifier, that simply assigns *all* proteins to the majority class, would have been 61.6%. However, this simple classifier is not useful at all for actual protein function annotation because the Recall for all other function classes (non-majority) will be 0. The *Text-kNN* classifier outperforms both of our baseline classifiers for *molecular function* classes. The evaluation results, namely the Precision, Recall, F-measure and MCC, for individual function classes are shown in Table 4.2.1. The highest value for each performance measure across the three classifiers is shown in bold.

When comparing the performance of *Text-kNN* to *Base-Prior*, we observe that the text-based classifier outperforms (with high statistical significance, $p < 0.05$) *Base-Prior* on almost all *molecular function* classes in all measures, except for the '*structural molecular activity'* class (GO:0005198). In our dataset, the majority of '*structural molecular activity'* proteins are annotated as '*binding'* as well. This means that the proteins that are associated with the *'structural molecular activity'* have similar representation as weighted term vectors as the proteins that are associated with the '*binding*' class. In our training dataset, proteins that are labeled as *'binding'* outnumber the proteins that are labeled as *'structural molecular activity'* by a factor of about 30. Therefore, during the evaluation, the ten nearest neighbours of proteins whose true label is *'structural molecular activity'* are often all proteins labeled as *'binding'*.

Consequently, most of the proteins belonging to the *'structural molecular activity'* class are only classified as *'binding'*, resulting in a lower Precision and Recall for the '*structural molecular activity'* class.

Molecular Function	**# Proteins**	**Text- KNN**				**Base-Prior**				**Base-Seq**			
		P	**R**	**F**	**M**	**P**	**R**	**F**	**M**	**P**	**R**	**F**	**M**
GO:0005488 binding	13400	0.65	**0.88**	**0.75**	**0.153**	0.63	0.64	0.63	0.000	**0.67**	0.75	0.71	0.117
GO:0003824 catalytic activity	3679	**0.52**	0.23	0.32	**0.241**	0.16	0.15	0.15	0.001	0.38	**00.29**	**0.33**	0.230
GO:0030528 transcription regulator activity	1595	0.44	0.24	0.31	0.290	0.07	0.07	0.07	0.001	**0.49**	**0.37**	**0.42**	**0.376**
GO:0005215 transporter activity	978	**0.59**	0.38	**0.46**	**0.450**	0.04	0.04	0.04	0.004	0.50	**0.43**	**0.46**	0.441
GO:0060089 molecular transducer activity	922	**0.39**	0.16	0.22	**0.248**	0.04	0.04	0.04	0.002	0.26	**0.27**	**0.27**	0.232
GO:0030234 enzyme regulator activity	606	**0.43**	0.05	0.08	**0.146**	0.03	0.03	0.03	0.009	0.16	**0.09**	**0.12**	0.112
GO:0005198 structural molecular activity	418	0.04	0.01	0.01	0.003	0.02	0.02	0.02	0.002	**0.11**	**0.11**	**0.11**	0.090
GO:0016247 channel regulator activity	72	**0.60**	**0.24**	**0.35**	**0.597**	0.01	0.01	0.01	0.003	0.00	0.00	0.00	0.000
GO:0009055 electron carrier activity	68	0.00	0.00	0.00	0.000	0.00	0.00	0.00	0.000	0.00	0.00	0.00	0.000
GO:0045182 translator regulator activity	26	0.00	0.00	0.00	0.000	0.00	0.00	0.00	0.000	0.00	0.00	0.00	0.000

Table 4.2.1: The performance of the text-based classifier, *Text-kNN*, over *molecular function* classes, compared with two baselines: *Base-Prior*, and *Base-Seq*. The column *#Proteins* shows the total number of proteins that are associated with each class in our dataset. The columns P, R, F, and M show the classifier's Precision, Recall, F-measure, and MCC respectively, over individual classes. Precision and Recall values of 0 for a class indicate that all the proteins belonging to that class are misclassified into another class.

In comparison to the *Base-Seq* classifier, our *Text-kNN* has a comparable (i.e. no statistically significant difference) – if not higher – Precision and MCC for most of the *molecular function* categories, but a lower Recall and F-measure for all but three classes. Notably, for the three *molecular function* classes that have fewer than 100 associated proteins, *Text-kNN* correctly classifies only one out of the three classes, while *Base-Seq* makes no correct classification at all.

4.2.2 Performance of *Text-kNN* on *biological process* classes

The overall accuracy measures of the classifiers for *biological process* classes are 17% for *Text-kNN*, 11% for *Base-Prior*, and 28% for *Base-Seq*. We note that the overall accuracy of an alternative hypothetical *Base-Prior* classifier that simply assigns *all* proteins to the majority class would have been 20%. However, as previously mentioned, such a classifier is not useful at all for the protein function prediction task. The performance for individual classes, measured using Precision, Recall, F-measure, and MCC are shown in Table 4.2.2. The highest value for each performance measure across the three classifiers is shown in bold

Biological Process	# Proteins	Text- KNN				Base-Prior				Base-Seq			
		P	R	F	M	P	R	F	M	P	R	F	M
GO:0065007 biological regulation	4532	0.23	**0.52**	0.31	0.065	0.20	0.24	0.22	0.003	**0.32**	0.48	**0.38**	**0.148**
GO:0032502 developmental process	4173	**0.22**	0.19	0.20	0.099	0.12	0.17	0.14	0.002	**0.22**	**0.24**	**0.23**	**0.143**
GO:0009987 cellular process	2237	0.24	**0.29**	0.26	0.103	0.17	0.14	0.15	0.002	**0.26**	0.27	**0.27**	**0.115**
GO:0050896 response to stimulus	2225	**0.25**	**0.16**	**0.19**	**0.116**	0.10	0.10	0.10	0.003	0.16	0.09	0.11	0.039
GO:0008152 metabolic process	2073	0.23	0.14	0.17	0.126	0.08	0.06	0.07	0.003	**0.28**	**0.34**	**0.31**	**0.236**
GO:0051234 establishment of localization	1505	0.32	0.20	0.25	0.209	0.05	0.05	0.05	0.002	**0.44**	**0.45**	**0.45**	**0.401**
GO:0016043 cellular component organization	1431	0.13	0.05	0.07	0.044	0.06	0.05	0.06	0.002	**0.15**	**0.12**	**0.13**	**0.090**

GO:0023052 signaling	1206	0.18	0.11	0.14	0.134	0.05	0.04	0.04	0.002	**0.30**	**0.28**	**0.29**	**0.240**
GO:0032501 multi-cellular organismal process	757	0.12	0.02	0.04	0.028	0.04	0.03	0.04	0.003	**0.24**	**0.11**	**0.16**	**0.148**
GO:0022414 reproductive process	432	**0.51**	**0.15**	**0.24**	**0.341**	0.02	0.02	0.02	0.001	0.14	0.03	0.05	0.062
GO:0051704 multi-organism process	340	**0.29**	**0.09**	**0.14**	**0.166**	0.01	0.01	0.01	0.001	0.09	0.04	0.05	0.051
GO:0040011 locomotion	212	0.13	0.01	0.01	0.002	0.01	0.01	0.01	0.001	**0.84**	**0.05**	**0.09**	**0.217**
GO:0040007 growth	206	**0.01**	**0.01**	**0.01**	**0.001**	**0.01**	**0.01**	**0.01**	0.001	0.00	0.00	0.00	0.000
GO:0051179 localization	189	**0.03**	**0.01**	**0.01**	**0.002**	0.01	**0.01**	**0.01**	0.001	0.00	0.00	0.00	0.000
GO:0022610 biological adhesion	160	**0.07**	**0.02**	**0.03**	**0.001**	0.01	0.01	0.01	0.001	0.00	0.00	0.00	0.000
GO:0008283 cell proliferation	147	**0.01**	**0.01**	**0.01**	**0.001**	**0.01**	**0.01**	**0.01**	0.001	0.00	0.00	0.00	0.000
GO:0000003 reproduction	120	0.00	0.00	0.00	**0.001**	**0.01**	**0.01**	**0.01**	0.001	0.00	0.00	0.00	0.000
GO:0002376 immune system response	93	**0.06**	**0.03**	**0.04**	**0.082**	0.00	0.00	0.00	0.000	0.00	0.00	0.00	0.000
GO:0016265 death	80	0.00	0.00	0.00	0.000	**0.01**	**0.01**	**0.01**	0.001	0.00	0.00	0.00	0.000
GO:0071554 cell wall organization	57	**0.38**	**0.08**	**0.13**	**0.207**	0.01	0.00	0.00	0.000	0.00	0.00	0.00	0.000
GO:0048511 rhythmic process	54	**0.31**	**0.06**	**0.10**	**0.002**	0.00	0.00	0.00	0.000	0.00	0.00	0.00	0.000
GO:0023046 signaling process	44	**0.01**	**0.01**	**0.01**	**0.001**	0.00	0.00	0.00	0.000	0.00	0.00	0.00	0.000
GO:0044085 cellular component biogenesis	20	**0.01**	**0.01**	**0.01**	**0.001**	0.00	0.00	0.00	0.000	0.00	0.00	0.00	0.000
GO:0043473 pigmentation	16	0.00	0.00	0.00	0.000	0.00	0.00	0.00	0.000	0.00	0.00	0.00	0.000

Table 4.2.2: The classification performance of the text-based classifier, *Text-kNN*, over *biological process* classes, compared with two baselines: *Base-Prior*, and *Base-Seq*. The column *#Proteins* shows the total number of proteins that are associated with each class in our dataset. The columns P, R, F, and M show the classifier's Precision, Recall, F-measure, and MCC respectively, over individual classes. Precision and Recall values of 0 for a class indicate that all the proteins belonging to that class are misclassified into another class.

For the *biological process* categories, we once again note that *Text-kNN* has a higher Precision, Recall and F-measure than *Base-Prior* on the majority of the classes. When comparing

Text-kNN to *Base-Seq*, we observe that *Base-Seq* has higher Precision and Recall on classes that have more than 200 associated proteins (11 out of the 24 classes). However, for process classes that have fewer than 200 associated proteins (with the exception of *"locomotion",* GO:0040011) *Base-Seq* does not make any correct classifications because it consistently misclassifies these proteins as belonging to the categories that have the largest number of associated proteins. Meanwhile, our text-based classifier is able to correctly classify the function of some of these proteins, albeit with low Precision and Recall.

4.3 Performance comparison between *Text-kNN* and *Text-SVM* classifiers

We compare the performance of *Text-kNN* with that of *Text-SVM*, which uses the LIBSVM (Chang and Lin, 2011) implementation of *Support Vector Machines* as the classifier. For *molecular function* classes, *Text-kNN* has a higher Precision but lower Recall, F-measure, and MCC than *Text-SVM* for the majority of the classes. For *biological process* classes, *Text-kNN* does not perform as well as *Text-SVM* on larger classes that have more than 200 associated proteins, but *Text-kNN* outperforms *Text-SVM* for smaller classes that have fewer than 200 associated proteins.

4.3.1 Performance of *Text-SVM* on *molecular function* classes

The overall classification accuracy of *Text-kNN* on *molecular function* classes is 62% while the classification accuracy of *Base-Seq* and *Text-SVM* is 58% and 39%. The results, namely the Precision, Recall, F-measure and MCC, for individual function classes are shown in Table 4.3.1. The highest value for each performance measure across the three classifiers is shown in bold.

When comparing the performance of *Text-kNN* to *Text-SVM* on *molecular function* classes, we observe that even though *Text-SVM* has a lower overall accuracy, it has significantly higher Recall, *F-measure* and *MCC* for 6 out of the 10 *molecular function* classes while *Text-kNN* has slightly higher Precision. In fact, the Recall rate of *Text-SVM* is about twice as high as that of

Molecular Function	**# Proteins**	**Text- KNN**				**Text-SVM**				**Base-Seq**			
		P	**R**	**F**	**M**	**P**	**R**	**F**	**M**	**P**	**R**	**F**	**M**
GO:0005488 binding	13400	0.65	**0.88**	**0.75**	0.15	**0.70**	0.73	0.71	**0.23**	0.67	**0.75**	0.71	0.12
GO:0003824 catalytic activity	3679	**0.52**	0.23	0.32	0.24	0.47	**0.46**	**0.46**	**0.36**	0.38	0.29	0.33	0.23
GO:0030528 transcription regulator activity	1595	0.44	0.24	0.31	0.29	0.43	**0.53**	**0.47**	**0.43**	**0.49**	0.37	0.42	0.38
GO:0005215 transporter activity	978	**0.59**	0.38	0.46	0.45	0.50	**0.55**	**0.52**	**0.50**	0.50	0.43	0.46	0.44
GO:0060089 molecular transducer activity	922	**0.39**	0.16	0.22	0.25	0.38	**0.33**	**0.35**	**0.33**	0.26	0.27	0.27	0.23
GO:0030234 enzyme regulator activity	606	**0.43**	0.05	0.08	0.15	0.33	**0.10**	**0.16**	**0.17**	0.16	0.09	0.12	0.11
GO:0005198 structural molecular activity	418	0.04	0.01	0.01	0.001	**0.11**	0.01	0.02	0.03	**0.11**	**0.11**	**0.11**	**0.09**
GO:0016247 channel regulator activity	72	**0.60**	**0.24**	**0.35**	**0.60**	0.50	0.06	0.11	0.18	0.00	0.00	0.00	0.00
GO:0009055 electron carrier activity	68	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
GO:0045182 translator regulator activity	26	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 4.3.1: The classification performance of the text-based classifiers *Text-SVM* over *molecular function* classes compared with *Text-kNN* and *Base-Seq*. The column *#Proteins* shows the total number of proteins that are associated with each class in our dataset. The columns P, R, F, and M show the classifier's Precision, Recall, F-measure, and MCC respectively, over individual classes. Precision and Recall values of 0 for a class indicate that all the proteins belonging to that class are misclassified into another class.

Text-kNN for five of the classes. Compared to *Base-Seq*, *Text-SVM* also has a higher Recall, F-Measure, and MCC for the majority of the classes.

For the *'binding'* class, *Text-SVM* has a higher Precision and a lower Recall than *Text-kNN*. This is probably because *'binding'* is the most abundant class and therefore, under the nearest-neighbour classification scheme, most of the proteins are assigned into the *'binding'* class, resulting in a higher Recall and a lower Precision. For classes that have fewer than 100 proteins in our dataset, both classifiers correctly classify only proteins in the *'channel regulator activity'* class (GO:0016247), where *Text-kNN* performs significantly better than *Text-SVM* according to all performance measures.

4.3.2 Performance of *Text-SVM* on *biological process* classes

The overall classification accuracy on *biological process* classes is 17% for *Text-kNN*, 26% for *Text-SVM* and 28% for the *Base-Seq* classifier. The evaluation results, namely the Precision, Recall, F-measure and MCC, for individual function classes are shown in Table 4.3.2. The highest value for each performance measure across the three classifiers is shown in bold.

For large *biological process* classes that have more than 200 proteins in our dataset (11 out of 24 classes), *Base-Seq* typically has the highest values according to all performance measures except for '*developmental process*' (GO:0032502), '*cellular process*' (GO:0009987), and '*response to stimulus*' (GO:0050896) where *Text-SVM* has higher value than *Base-Seq*. Compared to *Text-kNN*, *Text-SVM* has a higher Precision on all the large classes except for '*multi-organism process*' (GO:0051704). *Text-SVM* also has higher Recall, F-measure and MCC values than *Text-kNN* for 6 out of the 11 large *biological process* classes.

Biological Process	# Proteins	Text- KNN				Text-SVM				Base-Seq			
		P	R	F	M	P	R	F	M	P	R	F	M
GO:0065007 biological regulation	4532	0.23	**0.52**	0.31	0.07	0.24	0.50	0.33	0.09	**0.32**	0.48	**0.38**	**0.15**
GO:0032502 developmental process	4173	0.22	0.19	0.20	0.10	**0.27**	**0.35**	**0.31**	**0.21**	0.22	0.24	0.23	0.14
GO:0009987 cellular process	2237	0.24	0.29	0.26	0.10	**0.26**	**0.43**	**0.33**	**0.16**	**0.26**	0.27	0.27	0.12
GO:0050896 response to stimulus	2225	0.25	0.16	0.19	0.12	**0.31**	**0.22**	**0.26**	**0.19**	0.16	0.09	0.11	0.04
GO:0008152 metabolic process	2073	0.23	0.14	0.17	0.13	0.37	0.07	0.11	0.13	**0.28**	**0.34**	**0.31**	**0.24**
GO:0051234 establishment of localization	1505	0.32	0.20	0.25	0.21	0.39	0.26	0.31	0.29	**0.44**	**0.45**	**0.45**	**0.40**
GO:0016043 cellular component organization	1431	0.13	0.05	0.07	0.04	**0.17**	0.00	0.01	0.02	0.15	**0.12**	**0.13**	**0.09**
GO:0023052 signaling	1206	0.18	0.11	0.14	0.13	0.22	0.11	0.15	0.13	**0.30**	**0.28**	**0.29**	**0.24**
GO:0032501 multi-cellular organismal process	757	0.12	0.02	0.04	0.03	0.17	0.00	0.01	0.02	**0.24**	**0.11**	**0.16**	**0.15**
GO:0022414 reproductive process	432	0.51	**0.15**	**0.24**	**0.34**	**0.57**	0.09	0.15	0.22	0.14	0.03	0.05	0.06
GO:0051704 multi-organism process	340	**0.29**	**0.09**	**0.14**	**0.17**	0.20	0.01	0.01	0.04	0.09	0.04	0.05	0.05
GO:0040011 locomotion	212	0.13	0.01	0.01	0.00	0.00	0.00	0.00	0.00	**0.84**	**0.05**	**0.09**	**0.22**
GO:0040007 growth	206	**0.01**	**0.01**	**0.01**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
GO:0051179 localization	189	**0.03**	**0.01**	**0.01**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
GO:0022610 biological adhesion	160	0.07	**0.02**	**0.03**	0.00	**0.08**	0.01	0.02	**0.03**	0.00	0.00	0.00	0.00
GO:0008283 cell proliferation	147	**0.01**	**0.01**	**0.01**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
GO:0000003 reproduction	120	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
GO:0002376 immune system response	93	0.06	0.03	0.04	0.01	**0.25**	**0.04**	**0.06**	**0.09**	0.00	0.00	0.00	0.00

GO:0016265 death	80	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
GO:0071554 cell wall organization	57	**0.38**	**0.08**	**0.13**	**0.21**	0.25	0.03	0.05	0.09	0.00	0.00	0.00	0.00
GO:0048511 rhythmic process	54	**0.31**	**0.06**	**0.10**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
GO:0023046 signaling process	44	**0.01**	**0.01**	**0.01**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
GO:0044085 cellular component biogenesis	20	**0.01**	**0.01**	**0.01**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
GO:0043473 pigmentation	16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 4.3.2: The classification performance of the text-based classifiers *Text-SVM* over *biological process* classes compared with *Text-kNN* and *Base-Seq*. The column *#Proteins* shows the total number of proteins that are associated with each class in our dataset. The columns P, R, F, and M show the classifier's Precision, Recall, F-measure, and MCC respectively, over individual classes. Precision and Recall values of 0 for a class indicate that all the proteins belonging to that class are misclassified into another class.

For small *biological process* classes with fewer than 200 proteins in our dataset (13 out of 24 classes), all three classifiers perform poorly, with Precision and Recall that are lower than 0.10 for the majority of the classes. *Text-SVM* and *Base-Seq* do not classify any of the proteins correctly for the majority of small *biological process* classes. Therefore, *Text-kNN* typically performs slightly better according to all performance measures. The only exception is *'immune system process'* (GO:0002376) where *Text-SVM* has significantly higher Precision than *Text-kNN* and *Base-Seq.*

4.4 Performance evaluation on *textless* proteins

We present the performance of the *Text-kNN* and *Text-SVM* classifiers on proteins that have no associated abstracts listed in their UniProtKB/Swiss-Prot entries (*textless*). Table 4.4.1 shows the classification performance on textless proteins that are annotated with *molecular function* classes; Table 4.4.2 shows the performance on textless proteins that are annotated with

biological process. Classes that do not contain any textless proteins are not included in this evaluation. We note that some of the evaluated classes have a very small sample size of fewer than ten textless proteins available and therefore, the evaluation results for these classes are not statistically significant. As a point of reference, we also show the performance obtained in the cross-validation dataset on these same classes for proteins that have associated text. The results for the evaluated classes with textless proteins show Precision and Recall values that are consistent with those presented in Table 4.2.1 and Table 4.2.2 for proteins that have associated text.

4.4.1 Performance evaluation on *molecular function* classes for *textless* proteins

For *molecular function* classes, we observe that for the classes 'binding' (GO:0005488) and 'molecular transducer activity' (GO:0060089) the Precision obtained by both *Text-kNN* and *Text-SVM* when classifying textless proteins are higher than the average results obtained in the cross validation results for *Text-kNN.* As for the other classes, the Precision is only slightly lower. The only exception is the class '*transcription regulator activity*', which has only a single textless protein; the Precision here is very low (as quite a few textless proteins that belong to the '*binding*' class are misclassified into this class) while the Recall is 1.00.

Molecular Function	# Textless Proteins	Text-kNN (Textless)			Text-SVM (Textless)				*Text- KNN (Cross-validation)*		
		P	**R**	**F**	**P**	**R**	**F**		**P**	**R**	**F**
GO:0005488 binding	58	0.82	0.47	0.59	**0.88**	0.76	**0.81**		0.65	**0.88**	0.75
GO:0003824 catalytic activity	9	0.29	**0.56**	**0.38**	0.25	0.44	0.32		**0.52**	0.23	0.32
GO:0030528 transcription regulator activity	1	0.04	**1.00**	0.08	0.04	**1.00**	0.08		**0.44**	0.24	**0.31**
GO:0005215 transporter activity	5	0.50	0.20	0.29	0.50	0.20	0.29		**0.59**	**0.38**	**0.46**
GO:0060089 molecular transducer activity	7	**0.44**	0.57	**0.50**	0.38	0.43	0.40		0.39	0.16	0.22
GO:0005198 structural molecular activity	2	0.00	0.00	0.00	**0.00**	**0.00**	0.00		0.04	0.01	0.01

Table 4.4.1: Performance of the text-based classifiers over *molecular function* classes, for proteins that have no associated text. The columns P, R, F, and M show the classifier's Precision, Recall, F-measure, and MCC respectively, over individual classes. Precision and Recall values of 0 for a class indicate that all the proteins belonging to that class are misclassified into another class.

4.4.2 Performance evaluation on *biological process* classes for *textless* proteins

For *biological process* classes, the Precision and the Recall of *Text-kNN* for five out of the 13 evaluated classes are higher than the Precision and the Recall on the same classes for proteins that have associated text. For seven of the classes, the Precision and the Recall are both 0 on the textless dataset because all the proteins are misclassified into another class. However, we believe that this is only due to the small amount of textless proteins that are associated with these classes. Therefore, the results may not be reflective of how the classifier will perform on these classes for a larger dataset. When comparing the performance of *Text-SVM* to that of *Text-kNN* on the textless proteins, *Text-SVM* tends to have slightly lower Precision and Recall. The *Text-SVM* classifier also has a Precision and Recall of 0 for eight of the classes.

Biological Process	# Test Proteins	Text-kNN (Textless)			Text-SVM (Textless)				*Text- KNN (Cross-validation)*		
		P	**R**	**F**	**P**	**R**	**F**		**P**	**R**	**F**
GO:0065007 biological regulation	19	**0.28**	**0**.47	**0.35**	0.26	0.42	0.32		0.23	**0.52**	0.31
GO:0032502 developmental process	18	0.19	**0.22**	**0.21**	0.19	**0.22**	**0.21**		**0.22**	0.19	0.20
GO:0009987 cellular process	8	0.04	0.13	0.06	0.08	0.25	0.13		**0.24**	**0.29**	**0.26**
GO:0050896 response to stimulus	20	**0.38**	**0.30**	**0.33**	0.00	0.00	0.00		0.25	0.16	0.19
GO:0008152 metabolic process	7	**0.29**	**0.29**	**0.29**	0.00	0.00	0.00		0.23	0.14	0.17
GO:0051234 establishment of localization	9	**0.33**	**0.33**	**0.33**	0.25	0.22	0.24		0.32	0.20	0.25
GO:0016043 cellular component organization	6	0.00	0.00	0.00	0.00	0.00	0.00		**0.13**	**0.05**	**0.07**
GO:0023052 signaling	3	0.00	0.00	0.00	**0.20**	**0.33**	**0.25**		0.18	0.11	0.14
GO:0032501 multi-cellular organismal process	9	0.00	0.00	0.00	0.00	0.00	0.00		0.12	0.02	0.04
GO:0022414 reproductive process	7	0.00	0.00	0.00	0.00	0.00	0.00		**0.51**	**0.15**	**0.24**
GO:0051704 multi-organism process	1	0.00	0.00	0.00	0.00	0.00	0.00		0.00	0.00	0.00
GO:0040011 locomotion	3	0.00	0.00	0.00	0.00	0.00	0.00		0.00	0.00	0.00
GO:0002376 immune system response	1	0.00	0.00	0.00	0.00	0.00	0.00		0.00	0.00	0.00

Table 4.4.2: The classification performance of *Text-kNN* on *biological process* proteins that have no associated text is shown. The columns P, R, F, and M show the classifier's Precision, Recall, F-measure, and MCC respectively, over individual classes. Precision and Recall values of 0 for a class indicate that all the proteins belonging to that class are misclassified into another class.

4.5 Performance evaluation on the CAFA dataset

The prediction performance of our classifier on the CAFA dataset (which consists of 596 proteins) is compared to two other baseline classifiers chosen by the CAFA Challenge: *CAFA-Seq* and *Gotcha*. The results of this comparison are presented in this section. The results for *molecular function* classes are shown in Table 4.5.1, while the results for *biological process* classes are shown in Table 4.5.2. . Prediction performance was measured by CAFA using *Precision, Recall, and Specificity*. Precision and Recall are defined in section 4.1.3, whereas Specificity is calculated as:

$$\boldsymbol{Specificity} = \frac{\boldsymbol{TN}}{\boldsymbol{TN} + \boldsymbol{FP}} .$$

A Specificity value of 1 over a class indicates that all the proteins that are *not annotated* with that class label in the test dataset are correctly identified as such, and thus there are no false positives.

For *molecular function* classes, the results are shown at a confidence threshold of 0.95 for *Text-kNN, CAFA-Seq and GOtcha*. For *CAFA-Prior*, a confidence threshold of 0.01 is used, because at a confidence threshold of 0.02 *CAFA-Prior* makes no predictions for the *'transporter activity'* class, while at a confidence threshold of 0.14, *CAFA-Prior* makes no predictions for the *'catalytic activity'* class.

Function	**Text- KNN** (confidence = 0.95)			**CAFA-Prior** (confidence = 0.01)			**CAFA-Seq** (confidence = 0.95)			**GOtcha** (confidence = 0.95)		
	P	**R**	**S**	**P**	**R**	**S**	**P**	**R**	**S**	**P**	**R**	**S**
binding (212 proteins)	0.643	0.17	0.87	0.579	**1**	0.00	**0.9**	0.085	**0.987**	0.723	0.16	0.916
transporter activity (28 proteins)	0.00	0.00	0.97	0.077	**1**	0.00	0.5	0.036	**0.997**	**0.714**	0.179	0.994
catalytic activity (165 proteins)	0.312	0.03	0.95	0.451	**1**	0.00	0.714	0.03	**0.990**	**0.917**	0.067	0.**995**

Table 4.5.1 The text-based classifier, *Text-KNN*, is compared with baseline results provided by the CAFA challenge: *CAFA-Prior*, *CAFA-Seq*, and *GOtcha*. The confidence threshold used for each classifier is shown under its name in the respective column. The columns P, R, and S refer, respectively, to the Precision, Recall, and Specificity of the classifiers over individual classes. Precision and recall values of 0 for a class indicate that all the proteins belonging to that class are misclassified (when the confidence score is 0.95). *CAFA-Prior* always has a specificity value of 0, because it assigns all the proteins to each class, and as such the number of *true negatives* is always 0.

As shown in Table 4.5.1, the Precision of our classifier for two of the *molecular function* classes over the CAFA dataset is comparable to the results obtained through cross-validation (Table 4.3.1). Namely, at a confidence score threshold of 0.95, the Precision for the '*binding'* class was 0.64 on the cross-validation dataset and 0.74 on the CAFA dataset (which contains 212 'binding' proteins); the Precision for the '*catalytic activity'* class was 0.31 on the cross-validation targets and 0.49 on the CAFA targets (which contains 165 'catalaytic activity' proteins). In contrast, for the third class, *'transporter activity'* (28 proteins in the CAFA dataset), the Precision shown in Table 4.5.1 is 0. However, if we consider predictions made at a lower confidence threshold of 0.8, the Precision is 0.24 with a Recall of 0.18 compared with a Precision of 0.59 and a Recall of 0.38 on the cross-validation dataset. Notably, the *'transporter activity'* class is much larger in the cross-validation dataset with a total of 978 proteins as opposed to only 28 proteins in the CAFA dataset. When compared to the baseline classifiers, our text-based classifier has a significantly higher Precision than *CAFA-Prior* over the *'binding'* and *'transporter activity'* classes while the *CAFA-Seq* and *GOtcha* classifiers both have a higher Precision on all three

classes. (Again, we note that we report *CAFA-Prior*'s performance at a very low confidence level, because at a higher confidence threshold it makes no predictions for most classes.)

In terms of Recall, *CAFA-Prior* has a Recall of 1.0 on all classes at a confidence threshold of 0.01, but its Specificity is 0.0 because it assigns *every* GO category label to *each* protein (giving rise to 0 true negatives). *GOtcha* has the highest Recall for all three *molecular function* classes when compared to only *CAFA-Seq and Text-kNN*. Our classifier has a slightly higher Recall than *CAFA-Seq* on the *'binding'* class but a lower Recall on *'catalytic activity'* and '*transporter activity'.*

For *biological process* classes, only *CAFA-Seq* results are shown at a confidence threshold of 0.95. The results for our classifier, *Text-kNN,* are shown at a threshold of 0.75, while *GOtcha,* and *CAFA-Prior* results are shown at a threshold of 0.14 and 0.01 respectively. These thresholds are chosen because the classifiers make no prediction for over 75% of the classes at a higher confidence level.

As shown in Table 4.5.2, *CAFA-Seq* has the highest Precision on 11 out of the 14 classes. However, for the '*biological adhesion'* class, *CAFA-Seq, GOtcha and* our classifier did not make any correct predictions, that is, they all have Precision and Recall of 0. (Notably, *CAFA-prior* assigns all the proteins into each class, and as such by default always makes some correct predictions at a confidence threshold of 0.01). Moreover, both *CAFA-Seq* and our classifier have Precision and Recall of 0 for the *'multi-organism process'* class. Even though all three classifiers have a Precision and a Recall of 0 on these classes, the Specificity on those is still very close to 1. This is because the vast majority of proteins that belong to other classes are assigned to other classes, thus keeping true negatives correctly labeled as negatives.

Function	**Text- KNN** (*confidence = 0.75*)			**CAFA-Prior** (*confidence = 0.01*)			**CAFA-Seq** (*confidence = 0.95*)			**GOtcha** (*confidence = 0.14*)		
	P	**R**	**S**	**P**	**R**	**S**	**P**	**R**	**S**	**P**	**R**	**S**
biological regulation (114 proteins)	0.5	0.009	0**.997**	0.261	**1**	0	0**.632**	0.105	0.978	0.404	0.351	0.817
multi-organism process (29 proteins)	0.00	0.00	0.939	0.067	**1**	0	0.00	0.00	**0.99**	**0.286**	0.069	0.988
localization (60 proteins)	0.2	0.017	**0.989**	0.138	**1**	0	**0.44**	0.067	0.976	0.297	0.317	0.88
establishment of localization (38 proteins)	0.25	0.026	**0.992**	0.087	**1**	0	**0.5**	0.105	0.99	0.263	0.395	0.894
response to stimulus (106 proteins)	0.125	0.009	**0.979**	0.243	**1**	0	**0.5**	0.047	**0.985**	0.39	0.302	0.848
developmental process (83 proteins)	0.00	0.00	**0.997**	**0.19**	**1**	0	**0.556**	0.06	0.989	0.263	0.181	0.881
multicellular organismal process (87 proteins)	0.069	0.023	0.923	0.2	**1**	0	**0.625**	0.115	**0.983**	0.343	0.264	0.874
signalling (33 proteins)	**0.5**	0.03	**0.998**	0.076	**1**	0	0.25	0.061	0.985	0.077	0.061	0.94
biological adhesion (52 proteins)	0.00	0.00	0.971	0.06	**1**	0	0.00	0.00	**0.998**	0.00	0.00	0.993
cellular component organization (64 proteins)	0.00	0.00	**0.997**	0.147	**1**	0	**0.286**	0.031	0.987	0.192	0.156	0.887
cellular process (368 proteins)	0.857	0.016	**0.985**	0.844	**1**	0	**0.867**	0.071	0.941	0.866	0.829	0.309
metabolic process (213 proteins)	0.00	0.00	**0.991**	0.489	**1**	0	0.588	0.047	0.969	**0.633**	0.559	0.691
reproduction (25 proteins)	0.083	0.08	0.946	0.057	**1**	0	0.00	0.00	**0.995**	0.214	0.12	0.973
reproductive process (25 proteins)	0.083	0.08	0.946	0.057	**1**	0	0.00	0.00	**0.995**	**0.273**	0.12	0.981

Table. 4.5.2 - The text-based classifier, *Text-KNN*, is compared with baseline results provided by the CAFA challenge: *CAFA-Prior*, *CAFA-Seq*, and *GOtcha*. The confidence threshold used for each classifier is shown under its name in the respective column. The columns P, R, and S refer, respectively, to the Precision, Recall, and Specificity of the classifier over individual classes. Precision and recall values of 0 for a class indicate that all the proteins belonging to that class are misclassified (at the respective confidence level). *CAFA-Prior* always has a specificity value of 0, because it assigns all the proteins to each class, and as such the number of *true negatives* is always 0.

For the *'signaling'* class, our classifier has a significantly ($p < 0.05$) higher Precision than all other three classifiers, while for the *'binding'* class, our classifier has the second highest Precision after *CAFA-Seq*. Compared to *CAFA-Prior*, our classifier has a significantly ($p < 0.05$) higher Precision for four of the 14 classes and a slightly higher Precision for three of the 14 classes. (We note again though that this comparison is done where the confidence score for our classifier is 0.95 while for *CAFA-Prior* it is only 0.01. When both classifiers are compared at the 0.95 confidence level *CAFA-Prior* makes no predictions, and thus the text-based classifier vacuously outperforms it on all classes). In terms of Recall, *GOtcha* once again has the highest Recall (second to *CAFA-Prior* which has a Recall of 1) while both *CAFA-Seq* and our classifier demonstrate poor Recall on all the classes.

4.6 Summary of Results

In summary, the evaluation results show that our text-based classifiers, both *Text-kNN* and *Text-SVM,* consistently outperform the baseline classifier *Base-Prior*. When compared to *Base-Seq*, the text-based classifiers have higher Precision and Recall on the majority of *molecular function* classes but lower performance on the majority of the *biological process* classes. However, for *biological process* classes that have fewer than 200 associated proteins, *Base-Seq* does not make any correct predictions and, therefore, *Text-kNN* and *Text-SVM* have slightly higher performance according to all performance metrics.

We evaluated the ability of *Text-kNN* and *Text-SVM* to classify textless proteins by comparing their performance to the classification performance of *Text-kNN* on the cross-validation dataset. The results suggest that the text-based classifiers can classify textless proteins by utilizing the text features that are associated with homologous proteins. The evaluation results

showed comparable performance in both Precision and Recall - just as effectively as classifying proteins that have associated text.

The cross-validation results also indicate that the classification performance on *biological process* classes tends to be lower than the performance on *molecular function* classes, and that all three classifiers have poor performance on classes that have fewer than 200 proteins.

In the following Chapter, we conclude the thesis and propose ideas for future work that may improve the performance of our text-based function prediction system.

Chapter 5

Conclusion

In this thesis, we introduced a new function prediction system that uses text features as a basis for classifying proteins into function classes. We employed a previously presented strategy used in EpiLoc (Brady and Shatkay, 2008) for assigning text to proteins that have no associated text, enabling text-based function prediction for such proteins. We experimented with two different classifiers, *Support Vector Machines* (*Text-SVM*) and *k-Nearest Neighbour* (*Text-kNN*) and evaluated their performance. The performance of the two classifiers was evaluated using five-fold cross-validation on a dataset of 36,536 proteins and the results were compared to two other baseline classifiers.

5.1 Contributions

In this thesis we made the following contributions toward protein function prediction:

1. We introduced a new function prediction system that uses text features as a basis for classifying proteins into function classes. We employed a method for selecting text features from abstracts that was previously used for protein subcellular location prediction and applied it for the purpose of protein function prediction. The function classes were defined using *molecular function* and *biological processes* categories from the second level of the Gene Ontology. In order to handle proteins that lack associated text, we applied the method used by Brady and Shatkay (2008), in the task of protein subcellular location prediction, to assign text features to such proteins.
2. We evaluated the performance of our text-based system (*Text-kNN)* using stratified five-fold cross-validation over a dataset of 36,536 proteins, employing the standard

performance metrics of Precision, Recall, *F-measure* and *MCC*. We compared *Text-kNN* to two other baseline classifiers: *Base-Prior*, which assigns function classes based on the prior distribution of the classes in the dataset and *Base-Seq*, which assigns function classes based on sequence similarity between proteins. The results showed that our text-based prediction system has higher Precision *and* Recall than *Base-Prior* for both *molecular function* and *biological process* classes.

When compared to *Base-Seq*, *Text-kNN* showed higher Precision and but lower Recall for the majority of the *molecular function* classes. As for *biological process* classes, *Base-Seq* tends to have higher Precision and higher Recall than *Text-kNN*. The exceptions are the small classes that have fewer than 200 associated proteins and the classes *'reproductive process'* (GO:0022414), *'multi-organism process'* (GO:0051704), *'cell wall organization or biogenesis'* (GO:0071554), and *'rhythmic process'* (GO:0048511), for which *Text-kNN* has significantly higher Precision and Recall. The results from the CAFA challenge, shown in Section 4.5, also demonstrate that even though our text-based classifier, *Text-KNN* has a lower Recall than the sequence-based classifiers, it has significantly higher Precision than at least one of the sequence-based classifiers on the classes *'binding', 'signalling'* and *'biological regulation'.*

3. We implemented an alternative classifier for text-based prediction using Support Vector Machines (*Text-SVM*) and compared its performance to the previously mentioned *Text-kNN* and *Base-Seq*. The results show that *Text-SVM* typically has significantly higher Recall and F-measure but a lower Precision than *Text-kNN* for classifying *molecular function* classes. As for *biological process* classes, the results

showed that *Text-SVM* has higher Precision, Recall, and *F-measure* for the majority of the classes that have more than 200 associated proteins, while *Text-kNN* has higher performance for classes that have fewer than 200 associated proteins.

5.2 Future Work

The text-based protein function prediction system that we introduced in this thesis performed significantly better than the baseline classifier that assigns function classes based on class-distribution. It showed comparable, or even higher, Precision and Recall with respect to the sequence-based baseline classifier for several *molecular function* and *biological process* classes.

We note that from the onset, we view text as an important source of available information that can be used in conjunction with other types of protein features to improve function prediction. To this end, our cross-validation results and the evaluation results from the CAFA challenge suggest that the information obtained from text features and that obtained from sequence features complement each other, and the text-based and sequence-based classifiers have different strengths and weaknesses.

In Sections 4.2, 4.3, and 4.5, the results show that our text-based classifiers have significantly higher Precision and Recall than the sequence-based classifiers for several function classes (such as '*signaling'*, '*biological regulation'*, *'cell wall organization or biogenesis*', and '*rhythmic process'*). The text-based classifiers were also able to correctly classify proteins of small *biological process* classes when the sequence-based classifiers could not.

These results suggest that text features extracted from the biomedical literature contain information about protein function that is not readily evident in features obtained from protein sequences alone. As such, integrating text-based features with sequence-based – as well as with

other types of – features, will likely improve the performance of existing function prediction systems.

In our evaluation, we observed that the text-based classifiers do not perform as well for small classes that have fewer than 200 associated proteins. One possible reason for the lower level of performance in such cases is that the current feature selection method performs poorly in identifying meaningful characteristic text features due to the limited amount of associated text. Therefore, an immediate next step in this research is the evaluation of alternative statistics for selecting text features, in a way that would accommodate accurate classification even for function classes that only have a small number of associated proteins.

Currently, our system only uses the categories from the second level of the GO hierarchy as function classes due to the lack of proteins associated with the lower levels of GO. Therefore, once we improve feature selection for GO categories that have limited number of proteins, we would like to extend the system by using the categories at the lower levels of the GO hierarchy as function classes. This would support more specific description of the protein's function.

Another direction that we would like to explore in the future is the splitting of the dataset of proteins into *plants* and *animals* proteins, performing classification separately on the split dataset. This is motivated by the observation that some functions are only found in plants while others are only found in animal. Separating the dataset into *plants* and *animals* will also lead to a reduction in the total number of different function classes associated with each classifier. We believe that this splitting of the dataset will eliminate the mis-assignment of animal proteins into plant-specific functions and vice versa, thereby improving the Precision and Recall of our classifier.

References

Adams, M. D., & Sekelsky, J. J. 2002. From sequence to phenotype: reverse genetics in Drosophila melanogaster. *Nature Reviews Genetics, 3,* 189–198.

Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., & Walter, P. 2002. *Molecular Biology of the Cell* (4th ed.). New York: Garland Science.

Aloy, P., Ceulemans, H., Stark, A., Russell, R. B. 2003. The Relationship Between Sequence and Interaction Divergence in Proteins. *Journal of Molecular Biology., 332*(5), 989-998

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., & Lipman, D. J. 1990. Basic local alignment search tool. *Journal of Molecular Biology, 215*(3), 403-410.

Altschul, S. F., Madden T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Millwer, W., Lipman, D. J. 1997. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Research, 25(17), 3389–3402

Andrade, M. A., & Valencia, A. 1998. Automatic extraction of keywords from scientific text: application to the knowledge domain of protein families.*Bioinformatics, 14*(7), 600-607.

Apweiler, R., Bairoch, A., Wu, C. H., Barker, W. C., Boeckmann, B., Ferro, S., ... & Yeh, L. S. L. 2004. UniProt: the universal protein knowledgebase.*Nucleic Acids Research, 32*(suppl 1), D115-D119.

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., ... & Sherlock, G. 2000. Gene Ontology: tool for the unification of biology. *Nature genetics, 25*(1), 25.

Barrell, D., Dimmer, E., Huntley, R. P., Binns, D., O'Donovan, C., & Apweiler, R. 2009. The GOA database in 2009—an integrated Gene Ontology Annotation resource. *Nucleic Acids Research, 37*(suppl 1), D396-D403.

Bartlett, G. J., Annabel, E., & Thornton, J. M. 2005. Inferring protein function from structure. *Structural Bioinformatics, 44*, 387-407.

Bartlett, G., Borkakoti, N., & Thornton, J. 2003. Catalysing new reactions during evolution: Economy of residues and mechanism. *Journal of Molecular Biology, 331*, 829–860.

Benson, D. A., Boguski, M. S., Lipman, D. J., & Ostell, J. 1997. GenBank. *Nucleic Acids Research*, *25*(1), 1-6.

Bishop C. M. 2006. *Pattern Recognition and Machine Learnin*g (Vol. 4 No. 4.) New York: Springer.

Brady, S., & Shatkay, H. 2008. EpiLoc: a (working) text-based system for predicting protein subcellular location. In *Pacific Symposium of Biocomputing* (Vol. 13, pp. 604-615).

Brenner, S. E., Chothia, C., Hubbard, T. J., & Murzin, A. G. 1996. Understanding protein structure: using scop for fold interpretation. *Methods in Enzymology*, *266*, 635-643.

Briesemeister, S., Blum, T., Brady, S., Lam, Y., Kohlbacher, O., & Shatkay, H. 2009. SherLoc2: a high-accuracy hybrid method for predicting subcellular localization of proteins. *Journal of Proteome Research*, *8*(11), 5363-5366.

Camon, E., Magrane, M., Barrell, D., Lee, V., Dimmer, E., Maslen, J., ... & Apweiler, R. 2004. The Gene Ontology annotation (GOA) database: sharing knowledge in Uniprot with Gene Ontology. *Nucleic Acids Research*, *32*(suppl 1), D262-D266.

Carpenter, A. E., & Sabatini, D. M. 2004. Systematic genome-wide screens of gene function. *Nature Reviews Genetics*, *5*(1), 11-22.

Chang, C. C., & Lin, C. J. 2011. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, *2*(3), 27.

Chiang, J. H., & Yu, H. C. 2003. MeKE: Discovering the functions of gene products from biomedical literature via sentence alignment. *Bioinformatics, 19*(11), 1417–1422.

Chiang, J. H., Yu, H. C. & Hsu, H. J. 2004. GIS: A biomedical text-mining system for gene information discovery, *Bioinformatics*, *20*(1), 120–121

Chua, H. N., Sung, W. K., & Wong, L. 2006. Exploiting indirect neighbours and topological weight to predict protein function from protein–protein interactions. *Bioinformatics, 22*(13), 1623-1630.

Clark, W. T., & Radivojac, P. 2011. Analysis of protein function and its prediction from amino acid sequence. *Proteins: Structure, Function, and Bioinformatics*, *79*(7), 2086-2096.

Cohen, A. M., & Hersh, W. R. 2005. A survey of current work in biomedical text mining. *Briefings in Bioinformatics*, *6*(1), 57-71.

Conesa, A., Götz, S., García-Gómez, J. M., Terol, J., Talón, M., & Robles, M. 2005. Blast2GO: a universal tool for annotation, visualization and analysis in functional genomics research. *Bioinformatics*, *21*(18), 3674-3676.

Devore, J. L. 2011. *Probability and Statistics for Engineering and the Sciences*. Duxbury Press.

Feldmann, M., & Maini, R. N., (2003. TNF defined as a therapeutic target for rheumatoid arthritis and other autoimmune diseases. *Nature Medicine, 9*(10), 1245–1250.

Finn, R., Clements, J., & Eddy, S. 2011. HMMER web server: interactive sequence similarity searching. *Nucleic Acids Research, 39*(Web Server issue), W29-W37.

Fraser, A. G., Kamath, R. S., Zipperlen, P., Martinez-Campos, M., Sohrmann, M., & Ahringer, J. 2000. Functional genomic analysis of C. elegans chromosome I by systematic RNA interference. *Nature*, *408*(6810), 325-330.

Friedberg, I. 2006. Automated protein function prediction – the genomic challenge. *Briefings in Bioinformatics*, *7*(3), 225-242.

Gherardini, P.F., & Helmer-Citterich, M. 2008. Structure-based function prediction: approaches and applications. *Briefings in Functional Genomics and Proteomics*, *7*, 291–302.

Groth, D., Lehrach, H., & Hennig, S. 2004. GOblet: a platform for Gene Ontology annotation of anonymous sequence data. *Nucleic Acids Research*,*32*(suppl 2), W313-W317.

Groth, P., Weiss, B., Pohlenz, H. D., & Leser, U. 2008. Mining phenotypes for gene function prediction. *BMC Bioinformatics, 9*, 136.

Guan, Y., Myers, C. L., Hess, D. C., Barutcuoglu, Z., Caudy, A. A., & Troyanskaya, O. G. 2008. Predicting gene function in a hierarchical context with an ensemble of classifiers. *Genome Biology*, *9*(Suppl 1), S3.

Holm, L., Ouzounis, C., Sander, C., Tuparev, G., & Vriend, G. 1992. A database of protein structure families with common folding motifs. *Protein Science*, *1*(12), 1691-1698

Huang, Y. F., Chiu L. Y., Huang, C. C., & Huang C. K. 2010. Predicting RNA-binding residues from evolutionary information and sequence conservation. *BMC Genomics*, *11*(Suppl 4):S2

Huberts, D., & van der Klei, I. 2010. Moonlighting proteins: an intriguing mode of multitasking. *Biochimica Et Biophysica Acta*, *1803*(4), 520–525.

Izumitani, T., Taira, H., Kazawa, H., & Maeda, E. 2004, August. Assigning gene ontology categories (go) to yeast genes using text-based supervised learning methods. In *Computational Systems Bioinformatics Conference, 2004. CSB 2004. Proceedings. 2004 IEEE* (pp. 503-504. IEEE.

Jeffery, C. 1999. Moonlighting proteins. *Trends in Biochemical Sciences*, *24*(1), 8-11.

Jelkmann, W. 2007. Erythropoietin after a century of research: younger than ever. *European Journal of Haematology*, 78, 183–205.

Jensen, L. J., Gupta, R., Blom, N., Devos, D., Tamames, J., Kesmir, C., ... & Brunak, S. 2002. Prediction of human protein function from post-translational modifications and localization features. *Journal of Molecular Biology*, *319*(5), 1257-1266.

Jimenez-Sanchez, G., Childs, B., & Valle, D. 2001. Human disease genes. *Nature*, *409*(6822), 853-855.

Ko, S., & Lee, H. 2009. Integrative approaches to the prediction of protein functions based on the feature selection. *BMC Bioinformatics*, *10*(1), 455.

Koike, A., Niwa, Y., & Takagi, T. 2005. Automatic extraction of gene/protein biological functions from biomedical text. *Bioinformatics*, *21*(7), 1227-1236.

Lewis, D. 1998. Naive (Bayes) at forty: The independence assumption in information retrieval. *Machine Learning: ECML-98*, 4-15.

Lobley, A. E., Nugent, T., Orengo, C. A., & Jones, D. T. 2008. FFPred: an integrated feature-based function prediction server for vertebrate proteomes. *Nucleic Acids Research*, *36*(suppl 2), W297-W302.

Loewenstein, Y., Raimondo, D., Redfern, O., Watson, J., Frishman, D., Linial, M., Orengo, C., Thoront, J., & Tramontano, A. 2009. Protein function annotation by homology-based inference. *Genome Biology*, *10*(2), 207

Martin, D. M., Berriman, M., & Barton, G. J. 2004. GOtcha: a new method for prediction of protein function assessed by the annotation of seven genomes. *BMC Bioinformatics*, *5*, 178.

Marti-Renom, M. A., Ilyin, V. A., Sali, A. 2001. DBAli: a database of protein structure alignments. *Bioinformatics, 17*(8), 746-747.

Marti-Renom, M. A., Rossi, A., Al-Shahrour, F., Davis, F. P., Pieper, U., Dopazo, J., & Sali, A. 2007. The AnnoLite and AnnoLyze programs for comparative annotation of protein structures. *BMC Bioinformatics, 8*(Suppl 4), S4.

Matthews, B. W. 1975. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica Et Biophysica Acta, 405*(2), 442-451.

McManus, M. T., & Sharp, P. A. 2002. Gene silencing in mammals by small interfering RNAs. *Nature Reviews Genetics, 3*(10), 737–747.

Mewes, H. W., Ruepp, A., Theis, F., Rattei, T., Walter, M., Frishman, D., et al. 2011. MIPS: curated databases and comprehensive secondary data resources in 2010. *Nucleic Acids Research, 39*(suppl 1), D220-D224.

Mitchell, T. 1997. *Bayesian learning. Machine learning.* New York: McGraw-Hill Education.

Ng, P., & Henikoff, S. 2002. Accounting for Human Polymorphisms Predicted to Affect Protein Function. *Genome Research, 12*(3), 436-446

Pal, D., & Eisenberg, D. 2005. Inference of protein function from protein structure. *Structure, 13*(1), 121–130.

Pavlidis, P., Weston, J., Cai, J., & Grundy, W. N. 2001. Gene functional classification from heterogeneous data. In *Proceedings of the fifth annual international conference on Computational biology* (pp. 249-255. ACM.

Pazos, F., & Sternberg, M. 2004. Automated prediction of protein function and detection of functional sites from structure. *Proceedings of the National Academy of Sciences of the United States of America, 101*(41), 14754–14759.

Pellegrini, M. 2001. Computational methods for protein function analysis. *Current Opinion in Chemical Biology*, *5*(1), 46-50.

Peña-Castillo, L., Tasan, M., Myers, C. L., Lee, H., Joshi, T., Zhang, C., et al. 2008. A critical assessment of Mus musculus gene function prediction using integrated genomic evidence. *Genome Biology*, *9*(Suppl 1), S2.

Pérez, A. J., Perez-Iratxeta, C., Bork, P., Thode, G., & Andrade, M. A. 2004. Gene annotation from scientific literature using mappings between keyword systems. *Bioinformatics*, *20*(13), 2084-2091.

Petsko, G. A., & Ringe, D. 2004. *Protein structure and function*. Sinauer Associates Inc.

Porter, M. F. 2006. An algorithm for suffix stripping. *Program: electronic library and information systems*, *40*(3), 211-218.

Raychaudhuri, S., Chang, J. T., Sutphin, P. D., & Altman, R. B. 2002. Associating genes with gene ontology codes using a maximum entropy analysis of biomedical literature. *Genome Research, 12*(1), 203–214.

Redfern, O. C., Harrison A., Dallman, T., Pearl, F. M. & Orengo, C. A.. 2007. CATHEDRAL: a fast and effective algorithm to predict folds and domain boundaries from multidomain protein structures. *PLoS Computational Biology, 3*, E232.

Redfern, O. C., Harrison, A., Dallman, T., Pearl, F. M., & Orengo, C. A. 2007. CATHEDRAL: A Fast and Effective Algorithm to Predict Folds and Domain Boundaries from Multidomain Protein Structures. *PLoS Computational Biology*, *3*(11), e232.

Rigoutsos, I., Huynh, T., Floratos, A., Parida, L., & Platt, D. 2002. Dictionary-driven protein annotation. *Nucleic Acids Research, 30*, 3901–3916.

Rison, S. C., Hodgman, T. C., & Thornton, J. M. 2000. Comparison of functional annotation schemes for genomes. *Functional & Integrative Genomics*, *1*(1), 56-69.

Rost, B. 2002. Enzyme function less conserved than anticipated. *J Mol Biol*, *318*(2), 595–608.

Ruepp, A., Zollner, A., Maier, D., Albermann, K., Hani, J., Mokrejs, M., & Mewes, H. W. 2004. The FunCat, a functional annotation scheme for systematic classification of proteins from whole genomes. *Nucleic Acids Research*, *32*(18), 5539-5545.

Schwikowski, B., Uetz, P., & Fields, S. 2000. A network of protein-protein interactions in yeast. *Nature Biotechnology*, *18*(12), 1257-1261.

Shah, I., & Hunter, L. 1997. Predicting enzyme function from sequence: a systematic appraisal. *Proeedings of International Conference on Intelligent Systems for Molecular Biology*, *5*, 276-283.

Shah, P. K., Perez-Iratxeta, C., Bork, P., and Andrade, M. A. 2003. Information extraction from full text scientific articles: Where are the keywords? *BMC Bioinformatics*, 4: 20.

Shatkay, H., Höglund, A., Brady, S., Blum, T., Dönnes, P., & Kohlbacher, O. 2007. SherLoc: high-accuracy prediction of protein subcellular localization by integrating text and protein sequence data. *Bioinformatics*, *23*(11), 1410-1417.

Skolnick, J., & Brylinski, M. 2009. FINDSITE: a combined evolution/structure-based approach to protein function prediction. *Briefings in Bioinformatics*, *10*(4), 378-391.

Solomon E. P., Berg L. R., Martin D. W. 2002. Biology. (6th ed.). Thomson Coles Book.

Troyanskaya, O. G., Dolinski, K., Owen, A. B., Altman, R. B., Botstein, D. 2003. A Bayesian framework for combining heterogeneous data sources for gene function prediction (in Saccharomyces cerevisiae. *Proceedings of the National Academy of Sciences USA*, *100*, 8348–8353

UniProt Consortium. 2004. UniProt: the universal protein knowledgebase. *Nucleic Acids Research, 32*(Database issue), D115-D119.

Wass, M. N., & Sternberg, M. J. 2008. ConFunc—functional annotation in the twilight zone. *Bioinformatics*, *24*(6), 798-806.

Webb, A., Copsey, K., Cawley, G. 2011. *Statistical Pattern Recognition* (3rd ed.. Wiley.

Webb, E. C. 1992. *Enzyme nomenclature 1992. Recommendations of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology on the Nomenclature and Classification of Enzymes* (No. Ed. 6). Academic Press.

Whisstock, J. C., & Lesk, A. M. 2003. Prediction of protein function from protein sequence and structure. *Quarterly reviews of biophysics*, *36*(3), 307-340.

Wong, A., & Shatkay, H. 2011. Predicting Protein Function using Text Data from the Biomedical Literature. Extended abstract and oral presentation. *The Workshop of the Automated Protein Function SIG on Critical Assessment of Function Annotations.* ISMB, 2011.

Wong, A., & Shatkay, H. 2013. Protein Function Prediction using Text-based Features extracted from Biomedical Literature: The CAFA Challenge, 14(S3), S14.

Yang, Y., & Pedersen, J. O. 1997, July. A Comparative Study on Feature Selection in Text Categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning* (pp. 412-420). Morgan Kaufmann Publishers Inc.

Yao, Z., & Ruzzo, W. L. 2006. A regression-based K nearest neighbor algorithm for gene function prediction from heterogeneous data. *BMC Bioinformatics*. *7*(Suppl 1), S11.

Ye, Y., & Godzik, A. 2004. FATCAT: a web server for flexible structure comparison and structure similarity searching. *Nucleic Acids Research, 32*(suppl 2):W582-W585.

Zehetner, G. 2003. OntoBlast function: from sequence similarities directly to potential functional annotations by ontology terms. *Nucleic Acids Research 31*(13), 3799–3803.

Zhu, J., & Weng, Z. 2005 FAST: a novel protein structure alignment algorithm. *Proteins, 58*, 618-627.

Appendix A

List of Stop Words

a	be	each	herein	like	of	several	they	whenever
about	became	ec	hereupon	ltd	off	she	this	where
above	because	ed	her	made	often	should	thorough	whereafter
across	become	effected	herself	make	on	show	those	whereas
after	becomes	eg	him	many	only	showed	though	whereby
afterwards	becoming	either	himself	may	onto	shown	through	wherein
again	been	else	his	me	or	since	throughout	wherever
against	before	elsewhere	how	meanwhile	other	some	thru	whether
al	beforehand	enough	however	mg	others	somehow	thus	which
all	being	et	hr	might	otherwise	someone	to	while
almost	below	etc	ie	ml	our	something	together	whither
alone	beside	ever	if	mm	ours	sometimes	too	who
along	besides	every	ii	mo	ourselves	somewhere	toward	whoever
already	between	everyone	iii	more	out	still	towards	whom
also	beyond	everything	in	moreover	over	studied	try	whose
although	both	everywhere	inc	most	own	sub	type	why
always	but	except	incl	mostly	oz	such	ug	will
am	by	find	indeed	mr	per	take	under	with
among	came	for	into	much	perhaps	tell	unless	within
amongst	cannot	found	investigate	must	pm	th	until	without
an	cc	from	is	my	precede	than	up	wk
analyze	cm	further	it	myself	presently	that	upon	would
and	come	get	its	namely	previously	the	us	wt
another	compare	give	itself	neither	pt	their	used	yet
any	could	go	j	never	rather	them	using	you
anyhow	de	gov	jour	nevertheless	regarding	themselves	various	your
anything	dealing	had	journal	next	relate	then	very	yours
anywhere	department	has	just	no	s	thence	via	yourself
applicable	depend	have	kg	no one	said	there	was	yourselves
apply	did	he	last	nobody	same	thereafter	we	yr
are	discover	hence	latter	nor	seem	thereby	were	
around	dl	her	latterly	not	seemed	therefore	what	
as	do	here	lb	nothing	seeming	therein	whatever	
assume	does	hereafter	ld	now	seems	thereupon	when	
at	during	hereby	letter	nowhere	seriously	these	whence	

Table A.1: List of stop words that are removed from the abstracts during pre-processing

Appendix B

Code Repository

B.1 Retrieving Abstracts

B.1.1 Retriving abstracts from PubMed

Action:

- Retrieves abstract from PubMed based on list of PMIDs.

Location:

/fs/hs/projects/CAFA/pubmed/

Command:

perl getpmidfromweb.pl <*output directory*> <*pmid list*>

Options:

<*output directory*> - The path of the output directory where raw abstracts (HTML) will be saved.

<*pmid list*> - The path of a text file containing a list of PMIDs to retrieve (one PMID per line)

Data Files:

<*pmid list*> - /fs/hs/projects/CAFA/sws15/pmidlist.txt

B.1.2 Formatting raw HTML abstract files

Action:

- Formats the raw HTML abstracts retain only the text portion of abstract

Location:

/fs/hs/projects/CAFA/pubmed/

Command:

perl formatAbstract.pl <*input directory*> <*output directory*>

Options:

<*input directory*> - The path of the input directory where raw abstracts (HTML) were saved.

<*output directory*> - The path of the output directory where formatted abstracts will be saved.

B.1 Retrieving Abstracts

B.2.1 Creating .itame files

Action:

- Produces a single text file that contains all the individual abstracts

Location:

/fs/hs/projects/CAFA/itame/scripts

Command:

perl ITAMEfromID.pl <*abstracts directory*> <*output directory*>

Options:

<*abstracts directory*> - The path of the directory that contained the formatted abstracts

<*output directory*> - The path of the output directory where *abstracts.itame* will be created

B.2.2 Creating .stat files

Action:

- Preprocesses the abstracts by removing stop words, rare words, and applying Porter stemming.
- Creates a file in the same directory as the script that contains the word count for each individual abstract.

Location:

/fs/hs/projects/CAFA/itame/scripts/STATfromITAME

Command:

./GenTwoPass *[–s]* <*.itame file*> <*output file*>

Options:

[-s] – This flag indicates whether or not Porter stemming will be used

<*.itame file*> - The full path, including filename, of the .itame file created in B1.1.2

<*output file*> - The name of the output file that will be created in the same directory

Data File:

<*.itame file*> - /fs/hs/projects/CAFA/itame/output/itame/CAFA.itame

B.3 Selecting Text Features

Action:

- Uses the statistical test Z-Score to select *characteristic terms* to represent each GO class

Location:

/fs/hs/projects/CAFA/itame/scripts/CTfromSTAT

Command:

perl zsct.pl <*.id directory*> <*.stat file*> <*output directory*> ZSCORE <*z-score threshold*>

Options:

<*.id directory*> **-** The path of the directory that contains the .id files

<*.stat file*> **-** The full path, including the file name, to the .stat file generated in B1.2.2

<*output directory*> - The path of the directory where the output **ZSCharacteristicTerms.ct** file

<*z-score threshold*> - a numeric threshold for the Z-score test so that only terms that have a higher Z-score for n-1 classes are selected as characteristic terms

Data Files:

<.*id directory*> - /fs/hs/projects/CAFA/sws15/func-trimSwiss/newlvl1/sprotid/run*/fold*
OR - /fs/hs/projects/CAFA/sws15/proc-trimSwiss/newlvl1/sprotid/run*/fold*

<*.stat file*> - /fs/hs/projects/CAFA/itame/output/stat/CAFA.stat

B.4 Representing Proteins using Feature Vectors

B.4.1 Making Feature Vectors

Action:

- Creates vector representation of all the proteins in the dataset using the characteristic terms from feature selection.
- Each individual output file contains all the proteins for a specific GO class.
- Each new line in the output file represents a different protein.

Location:

/fs/hs/projects/CAFA/itame/scripts/VECTORfromCT

Command:

perl makeVectors.pl <*.id directory*> <*.stat file*> <*.ct file*> <*ignore list*> <*output directory*>

Options:

<*.id directory*> - The path of the directory that contains the .id files

<*.stat file*> - The full path, including the file name, to the .stat file generated in B1.2.2

<*.ct file*> - The full path, including the file name, to the .ct file generated in B1.

<*ignore list*> - A text file listing PMIDs that should be ignore

<*output directory*> - The path of the directory where the vector representation of each protein is created.

Data Files Location:

<.*id directory*> - /fs/hs/projects/CAFA/sws15/func-trimSwiss/newlvl1/sprotid/

<*.stat file*> - /fs/hs/projects/CAFA/itame/output/stat/

<.*ct file*> - /fs/hs/projects/CAFA/biopZS/ZSCharacteristicTerms.ct

B.4.2 Formatting Features Vectors for Matlab

Action:

- Formats the vector files generated in the previous section into the CSV format for compatibility with Matlab.

Location:

/fs/hs/projects/CAFA/itame/scripts/VECTORfromCT

Command:

perl mergeVectors.pl <*vector directory*>

Options:

<*vector directory*> - The path of the directory where the vector files were created. The vectors are reformatted to a comma-separated vector so that Matlab can read it.

Data File:

<*vector directory*> - /fs/hs/projects/CAFA/itame/output/molfZS/vec

OR - /fs/hs/projects/CAFA/itame/output/biopZS/vec

B.5 Running the classifiers on Matlab

Action:

- Runs the *text-knn* or *text-svm* classifier using the specified input

Location:

/fs/hs/projects/CAFA/itame/knnClassifier

Command:

1. Run ***matlab*** at command line.

2a. Run classifier using **textknn** <*goLeafFile*> <*vector directory*> <*expName*>

2b. Run classifier using **textsvm** <*goLeafFile*> <*vectorFolder*> <*expName*>

Options:

<*goLeafFile*> - The path to the goLeaf file. The file list the GO classes and # proteins per class. (**Default**: ./goLeaf1above15.txt (mol func) **OR** ./goLeaf1pabove15.txt (bio proc)

<*vectorFolder*> **-** The path of the directory where the vector files were created. The vectors are reformatted to a comma-separated vector so that Matlab can read it.

<*expName*> - The name of the experiment is added to the output files as a prefix.

Output:

expName_precision.csv **–** Contains the precision values of each class for individual runs.

expName_recall.csv – Contains the recall values of each class for individual runs.

expName_ConfMat#.csv – Contains the confusion matrix for the # run.

expName_avgP.csv – Contains the average precision values of each class across all runs.

expName_avgR.csv –Contains the average recall values of each class across all runs.

expName_avgConfMat.csv – Contains the average confusion matrix across all runs.